

OGC Testbed-15
Catalogue and Discovery Engineering Report

Table of Contents

1. Subject	4
2. Executive Summary	5
2.1. Problem Statement	5
2.2. Requirements	5
2.3. Achievements	6
2.3.1. Key Use Cases	6
2.4. Findings and Recommendations	7
2.4.1. Alignment with the OGC API - Features standard	7
2.4.2. Offerings defined in OGC 14-055r2	8
2.4.3. OpenSearch	8
2.5. Document contributor contact points	8
2.6. Foreword	8
3. References	10
3.1. Normative references	10
3.2. Other references	12
4. Terms and definitions	13
4.1. Abbreviated terms	14
4.2. Symbols	15
4.2.1. JSON Schema diagrams	15
4.2.2. JSONPath	15
4.2.3. Unified Modeling Language	15
4.3. Data dictionary tables	15
5. Overview	16
6. Catalogue Service Specification	17
6.1. Introduction	17
6.2. Service Interface	17
6.2.1. OpenSearch Description Document	17
6.2.2. Facet Discovery	20
6.3. Service Management Interface	24
6.3.1. Landing Page	26
6.3.2. API definition	28
6.3.3. Conformance statements	30
6.3.4. Collections	31
6.3.5. Services	32
6.3.6. Brief, summary responses	35
7. Data Model	37
7.1. GeoJSON EOPAD Metadata Encoding	37
7.1.1. Design Approach	37

7.1.2. Requirements class: Feature	37
Properties	41
7.1.3. Requirements class: Metadata Information	43
7.1.4. Requirements class: Service Identification	44
ServiceContact	47
Attribution	51
Agent	52
ProvenanceStatement	54
Activity	55
Entity	56
Association	58
Plan	58
Standard	59
CompoundPriceSpecification	61
7.1.5. Requirements class: Rights Information	62
LicenseDocument	64
RightsStatement	64
Use Constraints	65
Access Constraints	66
7.1.6. Requirements class: Descriptive Keywords	66
Category	70
Concept	71
7.1.7. Requirements class: Related URL	71
Links	73
Link	76
Offering	80
Operation	88
7.1.8. Requirements class: Spatial Information	90
Spatial Extent	90
7.1.9. Requirements class: Acquisition Information	91
7.2. GeoJSON Search Response Encoding	92
7.2.1. Requirements class: FeatureCollection	92
FacetedResults	96
DataSource	99
Facet	100
Term	102
8. Software Design	105
8.1. Software static architecture	105
8.2. Interfaces context	106
8.3. Software components design	107
8.3.1. D117 - Catalog Server (Deimos)	107

Metadata published	113
Interoperability tests	114
Lessons Learned	114
8.3.2. D121 - Catalog Server (GMU)	115
Description	115
Interfaces implemented	115
Metadata published	116
Interoperability tests	124
Lessons learned	124
8.3.3. D122 - Client (Deimos)	124
Description	124
8.3.4. D123 - Client (Compusult)	135
Searching External Catalogs	135
Service Management Interface	137
Service Chaining	139
Interoperability tests	140
Lessons learned	140
8.3.5. D118 - WPS Server (52°North)	141
Processes published	141
Catalog integration	142
Interoperability tests	143
8.3.6. D124 - WPS Server (Compusult)	143
Process Wrapper	143
Interoperability tests	144
8.3.7. DataBio Catalog Server (Spacebel)	144
Description	144
Interfaces implemented	144
Metadata published	150
Interoperability tests	151
Lessons learned	151
Future work	151
9. Arctic Discovery Catalogue	152
9.1. Overview	152
9.2. Architecture	152
9.3. Training CSW	153
9.4. Evergreen Catalogue	154
9.5. WES Sync	156
Appendix A: Abstract Test Suite	158
Appendix B: Interpreting JSON as JSON-LD	159
B.1. Introduction	159
B.1.1. JSON-LD	159

B.1.2. Semantic Web and JSON-LD	161
B.1.3. JSON-LD and Structured Data	161
B.2. JSON-LD @context definition	161
B.2.1. JSON-LD 1.1	161
B.2.2. Relation with DCAT and GeoDCAT-AP	168
B.2.3. Relation with Schema.org	168
Appendix C: Encoding Examples	172
C.1. Service Interfaces	172
C.1.1. OpenSearch Description Document	172
C.1.2. OpenSearch Response	172
C.1.3. OpenAPI Description	172
C.2. Metadata Examples	188
C.2.1. Example 1: MultiSensorNDVI Application	188
GeoJSON	188
JSON-LD (Compacted)	192
JSON-LD (Expanded)	192
JSON-LD (Expanded - schema.org)	196
C.2.2. Example 2: NDVI Calculation	197
Application Package (DeployProcess)	197
GeoJSON	201
JSON-LD (Expanded)	207
C.2.3. Example 3: Cloud coverage for an area of interest	209
Application Package (DeployProcess)	209
Appendix D: JSON Schema Documents	212
D.1. Schemas for EOPAD JSON validation	212
D.2. Schemas for Faceted Results validation	235
D.2.1. Definitions/Query (OGC 17-047)	235
D.2.2. Definitions/Facet	237
D.2.3. Definitions/DataSource	238
D.2.4. Definitions/FacetedResults	238
D.3. Schemas for Explain Document validation	239
Appendix E: OGC API - Processes, Transactional Profile OpenAPI 3.0 specification	245
Appendix F: Revision History	249

Publication Date: 2019-12-12

Approval Date: 2019-11-22

Submission Date: 2019-10-15

Reference number of this document: OGC 19-020r1

Reference URL for this document: <http://www.opengis.net/doc/PER/t15-D010>

Category: OGC Public Engineering Report

Editor: Yves Coene

Title: OGC Testbed-15: Catalogue and Discovery Engineering Report

OGC Public Engineering Report

COPYRIGHT

Copyright © 2019 Open Geospatial Consortium. To obtain additional rights of use, visit <http://www.opengeospatial.org/>

WARNING

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Public Engineering Report should not be referenced as required or mandatory technology in procurements. However, the discussions in this document could very well lead to the definition of an OGC Standard.

LICENSE AGREEMENT

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD. THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to

indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Chapter 1. Subject

This OGC Testbed-15 Engineering Report (ER) describes the results of the Earth Observation (EO) Process and Application (EOPAD) Task in the Cloud Processing and Portrayal (CPP) thread of OGC Testbed-15. The ER presents the data model and service interface of the catalogue service allowing for discovery of EO applications and related processing services for subsequent deployment and/or invocation in a distributed environment.

The ER also provides the architectural and implementation details of the software components that were developed as part of the activity and which interact through the described data model. These software components include catalogue clients, catalogue servers and transactional Web Processing Service (WPS-T) servers.

Chapter 2. Executive Summary

2.1. Problem Statement

Several platforms have emerged that provide access to Earth Observation data and processing capacities. These platforms host very large datasets, which makes a paradigm shift from data download and local processing towards application upload and processing close to the physical location of the data more and more important. To interpret peta- and exascale scientific data, capabilities of these platforms need to be combined in future.

OGC Testbed-13 and Testbed-14 Engineering Reports proposed solutions for packaging, deployment and execution of applications in cloud environments that expose standardized interfaces such as the OGC Web Processing Service (WPS). As long as a dedicated standardized interface such as an OGC WPS instance, a container execution environment (e.g. Docker), and data access are provided, the proposed approach was agnostic to the target cloud platform. In order for end-users to exploit such applications and already deployed processing capacities, facilities must be provided for users to discover the applications and to be provided with detailed descriptions and invocation instructions. This includes both applications to be deployed as well as already deployed applications that are available behind WPS interfaces.

The purpose of the current ER is to define the building blocks through which such applications and related services can be exposed through a Catalogue service:

- Data model,
- Service interface,
- Service management interface.

2.2. Requirements

The following requirements were provided as input for the Catalogue and Discovery activity:

The **Data Model** should provide:

- A unique identifier for each application or service.
- Descriptive information such as title/abstract etc.
- Classification information that could be used for faceted search / browse.
- Descriptions of the data types / constraints of the processing inputs and outputs. These should be formally described such that machine-decisions can be made regarding suitability of input data and pipelining of tool workflows.
- Accuracy / error information regarding the algorithms.
- Version information and associated application provenance.
- Interactive or non-interactive.
- Execution environment constraints (e.g. Docker version, hardware requirements) or Execution end-points (e.g. URLs).

- Metrics for costs estimation (data sizing, processing costs).
- Any other element as mutually agreed upon at the kick-off meeting.

The **Service Interface** should provide the following capabilities:

- Search interface for discovery by facet and free-text in combination.
- Request interface for retrieval of facet dictionary to support browse.
- Request interface for retrieval of Application detailed metadata, e.g. for display of an application landing page in catalogue web User Interface (UI).
- Management interface for the Creating, Update and Deletion of records in the catalogue.

2.3. Achievements

The bindings for the Catalogue and GeoJSON Data Model proposed in this ER are consistent with existing OGC Standards related to OWS Context and OGC Extensions of OpenSearch, namely [\[OGC 10-032r8\]](#), [\[OGC 13-026r9\]](#), [\[OGC14-055r2\]](#) and [\[OGC17-047\]](#). Support for facet discovery and faceted search responses was borrowed from existing OASIS SRU specifications and the [SRU extension of OpenSearch](#).

The proposed Data Model relies on OGC OWS Context [\[OGC14-055r2\]](#) Offerings to describe service or application access mechanisms and endpoints. Additional offering types were needed to represent services based on the OGC API - Common draft specification. These no longer have the concept of "operation" and applications accessible for deployment as Docker images. The Application Package defined in Testbed-14 is included in the application metadata as a "WPS DeployProcess" offering. In this way, inputs and outputs of the processing are formally described as part of the processDescription which is part of the Application Package. This allows for input/output compatibility checks during process workflow modelling.

In addition to the GeoJSON-based model, the corresponding JSON-LD representation is proposed as well in this ER. A service or application described in the catalog is modelled as a dcat:DataService in [\[DCAT-2\]](#). This allows for quality related information to be associated using the Data Quality Vocabulary [\[DQV\]](#). Where possible, the Data Model was aligned with [\[GeoDCAT-AP\]](#).

For the required transactional capabilities of the Catalogue interface, the OpenAPI Specification (formerly Swagger Specification, is an API description format for REST APIs) and the emerging OGC API - Common draft specification were considered.

The different interfaces described in this Engineering Report were implemented by multiple organizations and interoperability tests between client and server component implementations from participating organizations were successfully performed.

2.3.1. Key Use Cases

The following key use cases were successfully covered:

Use Case A: "As owner of an application (still to be deployed or already deployed) or operator of a service, I shall be able to register my application in the Catalogue Service. I shall be able to update the metadata about the application stored in the Catalogue. Finally, at the end of life of the

application, I shall be able to remove its metadata from the catalogue".

Use Case B.1: "As owner of an application, I shall be able to deploy the application on a cloud platform, using for instance an Execution Management Service (EMS) as defined in Testbed-14. After the deployment, the cloud platform shall be able to register the application information in the Catalogue Service via a machine-to-machine interface".

Use Case B.2: "As owner of an application, I shall be able to un-deploy the application from the cloud platform which shall therefore delete the metadata record relative to the application from the Catalogue Service via a machine-to-machine interface".

Use Case C: "As a user of applications, I shall be able to use the Catalogue Service to find an application to perform a specific processing algorithm such as a normalized difference vegetation index (NDVI). I shall be able to fill a form with the available search criteria or uses a free text form to specify some keywords and then the Catalogue service shall return all the applications which match the specified criteria. The interaction between the user and the Catalogue Service shall be mediated by a client of the service which shall provide a GUI to ease the user experience".

Use Case D: "As a user of applications, I shall be able to use the Catalogue Service to first find an application to perform a processing algorithm (e.g. classification) and then another application which can be fed with the output of the previous one (e.g. change detection) in order to create a complex application (a workflow or chain) to run systematically over a certain area of interest. The interaction between the user and the Catalogue Service or the cloud platform on the other hand shall be mediated by a client of the Catalogue service which provides a GUI to ease the user experience. The client shall support the user in verifying that the two applications are compatible (i.e. the input type of the second correspond to the output of the first) thanks to the information returned by the Catalogue Service".

2.4. Findings and Recommendations

2.4.1. Alignment with the OGC API - Features standard

The proposed OpenSearch Catalogue binding and GeoJSON Data Model with OGC API - Features - Part 1: Core [\[OGC17-069r2\]](#) and the draft OGC API - Common cannot fully be harmonized due to a number of incompatibilities:

- The Testbed-15 team supports the recommendation made in the OGC API Hackaton Engineering report OGC 19-062 to align the time and bbox related search parameters from the OGC OpenSearch Geo and Time Extensions (OGC 10-032r8) and the OGC API - Features - Part 1: Core standard.
- Also, additional work is needed to harmonise the link model and OpenSearch response metadata in GeoJSON (OGC 17-047) with the OGC API Features Standard. This may require an update to OGC 14-055r2 (which defines a link model incompatible with OGC API Features) and the specifications that are based on the OGC 14-055r2 specification, including Earth Observation Product (OGC 17-003), Earth Observation Collection (OGC 17-084) and OpenSearch Response (OGC 17-047).

2.4.2. Offerings defined in OGC 14-055r2

The Testbed-15 team recommends extending the offering types defined in [OGC14-055r2] with additional offerings such as a Docker image, or resources defined in the draft OGC API - Processes draft specification and OGC API - Features standard which were used in the activities described in this ER.

2.4.3. OpenSearch

There is currently no search parameter defined in [OGC13-026r9] supporting the retrieval of features (e.g. services/applications, EO collections, EO products) based on an available offering. In other words, return all applications available as Docker images. Similarly, there is no search parameter defined in OpenSearch Discovery [OGC13-026r9] to retrieve resources of a specific kind, which may be useful if a single resource catalog includes different kinds of features, such as EO services/applications, EO collections and EO products.

While it is currently possible to distinguish the URL templates in a single OpenSearch Description Document (OSDD) to be used for searching "products" and "collections" by their "rel" attribute ("results" or "collection" respectively), there is currently no agreed "rel" attribute value for searching "services" or "applications".

2.5. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Contacts

Name	Organization	Role
Yves Coene	Spacebel s.a.	Editor
Ken Geange	Compusult	Contributor
Mark Lawrence	Compusult	Contributor
Christian Autermann	52°North GmbH	Contributor
Benjamin Proß	52°North GmbH	Contributor
Koushik Panda	Deimos	Contributor
David Cordeiro	Deimos	Contributor
Eugene Genong Yu	George Mason University	Contributor

2.6. Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide

supporting documentation.

Chapter 3. References

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

3.1. Normative references

- IETF: RFC 7159, The JavaScript Object Notation (JSON) Data Interchange Format, March 2014 [<http://www.ietf.org/rfc/rfc7159.txt>]
- IETF: RFC 7946, The GeoJSON Format [<https://tools.ietf.org/html/rfc7946>]
- DCMI: Dublin Core DCMI Metadata Terms, Dublin Core Metadata Initiative, 14/06/2012 [<http://www.dublincore.org/specifications/dublin-core/dcmi-terms/>]
- W3C: Data Catalog Vocabulary (DCAT), W3C Recommendation 16 January 2014 [<https://www.w3.org/TR/vocab-dcat/>]
- W3C: Linked Data Platform 1.0, W3C Recommendation 26 February 2015 [<https://www.w3.org/TR/ldp/>]
- W3C: JSON-LD 1.0, A JSON-based Serialization for Linked Data, W3C Recommendation 16 January 2014 [<https://www.w3.org/TR/json-ld/>]
- W3C: JSON-LD 1.0 Processing Algorithms and API, W3C Recommendation 16 January 2014 [<https://www.w3.org/TR/json-ld-api/>]
- W3C: PROV-O: The PROV Ontology, W3C Recommendation 30 April 2013 [<https://www.w3.org/TR/prov-o/>]
- W3C: Data on the Web Best Practices: Data Quality Vocabulary, W3C Working Group Note 15 December 2016 [<https://www.w3.org/TR/vocab-dqv/>]
- W3C: RDF 1.1 XML Syntax, W3C Recommendation 25 February 2014 [<http://www.w3.org/TR/rdf-syntax-grammar/>]
- W3C: RDF 1.1 Turtle, Terse RDF Triple Language, W3C Recommendation 25 February 2014 [<http://www.w3.org/TR/turtle/>]
- OASIS: OASIS - searchRetrieve: Part 3. APD Binding for SRU 2.0 Version 1.0, OASIS Standard, 30 January 2013 [<http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/os/part3-sru2.0/searchRetrieve-v1.0-os-part3-sru2.0.html>]
- OASIS: OASIS - searchRetrieve: Part 4. APD Binding for OpenSearch Version 1.0, OASIS Standard, 30 January 2013 [<http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/searchRetrieve-v1.0-part4-opensearch.html>]
- OASIS: OASIS - searchRetrieve: Part 7. Explain Version 1.0, OASIS Standard, 30 January 2013 [<http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/searchRetrieve-v1.0-part7-explain.html>]
- OGC: OGC 06-121r9, OWS Common Implementation Standard [https://portal.opengeospatial.org/files/?artifact_id=38867&version=2]
- OGC: OGC 10-032r8, OGC® OpenSearch Geo and Time Extensions [<https://portal.opengeospatial.org/>]

files/?artifact_id=56866]

- OGC: OGC 11-052r4, OGC® GeoSPARQL - A Geographic Query Language for RDF Data [https://portal.opengeospatial.org/files/?artifact_id=47664]
- OGC: OGC 11-035r1, EO Product Collection, Service and Sensor Discovery using the CS-W ebRIM Catalog [https://portal.opengeospatial.org/files/?artifact_id=53276]
- OGC: OGC 13-026r9, OGC® OpenSearch Extension for Earth Observation (OpenSearch-EO) [<http://docs.opengeospatial.org/is/13-026r8/13-026r8.html>]
- OGC: OGC 13-068, OGC® OpenSearch Extension for Correlated Search [https://portal.opengeospatial.org/files/?artifact_id=55219]
- OGC: OGC 14-055r2, OGC® OWS Context GeoJSON Encoding Standard [<http://docs.opengeospatial.org/is/14-055r2/14-055r2.html>]
- OGC: OGC 17-047, OGC® OpenSearch-EO GeoJSON(-LD) Response Encoding Standard [https://portal.opengeospatial.org/files/?artifact_id=83328] (To be published soon)
- OGC: OGC 17-069r3, OGC API - Features - Part 1: Core [<http://docs.opengeospatial.org/is/17-069r3/17-069r3.html>]
- OGC: OGC 17-084, EO Collection GeoJSON(-LD) Encoding Standard [https://portal.opengeospatial.org/files/?artifact_id=83226]
- OGC: OGC 18-001r1, GeoDCAT-AP Discussion Paper [https://portal.opengeospatial.org/files/?artifact_id=82475]
- OGC: OGC 18-049r1, OGC Testbed-14: Application Package Engineering Report [<https://docs.opengeospatial.org/per/18-049r1.html>]
- OGC: OGC 18-050r1, OGC Testbed-14: ADES & EMS Results and Best Practices Engineering Report [<https://docs.opengeospatial.org/per/18-050r1.html>]
- OGC: OGC 19-062, OGC API Hackathon 2019 Engineering Report [https://portal.opengeospatial.org/files/?artifact_id=89634] (To be published soon)
- OAI: OpenAPI Specification, Version 3.0.2, OpenAPI Initiative [<https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.2.md>]
- OpenSearch community: OpenSearch 1.1 Draft 5 [<http://www.opensearch.org/Specifications/OpenSearch/1.1>]
- OpenSearch community: OpenSearch Parameter Extension, Version 1.0 Draft 2 [http://www.opensearch.org/Specifications/OpenSearch/Extensions/Parameter/1.0/Draft_2]
- OpenSearch community: OpenSearch SRU Extension, Version 1.0 Draft 1 [http://www.opensearch.org/Community/Proposal/Specifications/OpenSearch/Extensions/SRU/1.0/Draft_1]
- OpenSearch community: OpenSearch Semantic Extension, Version 1.0 Draft 1 [http://www.opensearch.org/Community/Proposal/Specifications/OpenSearch/Extensions/Semantic/1.0/Draft_1]
- EC: DCAT-AP Version 1.2, European Commission [<https://joinup.ec.europa.eu/solution/dcat-application-profile-data-portals-europe/releases>]
- EC: GeoDCAT-AP Version 1.0.1, European Commission [<https://joinup.ec.europa.eu/solution/geodcat-application-profile-data-portals-europe/distribution/geodcat-ap-101-pdf>]
- ESDIS: Unified Metadata Model Services (UMM-S), Revision 1.2, Earth Science Data and

https://wiki.earthdata.nasa.gov/download/attachments/49448405/UMM-S_V1.2_20180530.docx?version=1&modificationDate=1528375678575&api=v2]

- Schema.org Community Group: Schema.org [<https://schema.org/>]

3.2. Other references

- W3C: JSON-LD 1.1, A JSON-based Serialization for Linked Data, W3C Working Draft 10 May 2019 [<https://www.w3.org/TR/json-ld11/>]
- W3C: Data Catalog Vocabulary (DCAT) - Version 2, W3C Candidate Recommendation 03 October 2019 [<https://www.w3.org/TR/2019/CR-vocab-dcat-2-20191003/>]
- W3C: Data Catalog Vocabulary (DCAT) - Version 2 - W3C Working Draft 28 May 2019 [<https://www.w3.org/TR/vocab-dcat-2/>]
- JSONPath [<http://goessner.net/articles/JsonPath/index.html>]
- JSON Schema [<https://tools.ietf.org/html/draft-zyp-json-schema-04>]
- OGC: OGC API - Common draft specification [https://github.com/opengeospatial/oapi_common]
- Negotiating Profiles in HTTP, draft-svensson-accept-profile-00 [<https://profilenegotiation.github.io/I-D-Accept%2D-Schema/I-D-accept-schema>]

Chapter 4. Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [\[OGC06-121r9\]](#) shall apply. In addition, the following terms and definitions apply.

- **compaction**

While expansion removes context from a given input, compaction's primary function is to perform the opposite operation: to express a given input according to a particular context. In JSON-LD, compaction applies a context that specifically tailors the way information is expressed for a particular person or application. This simplifies applications that consume JSON or JSON-LD by expressing the data in application-specific terms, and it makes the data easier to read by humans [\[JSON-LD 1.0 Processing Algorithms and API\]](#).

- **context**

A set of rules for interpreting a JSON-LD document as specified in the section "The Context" of the JSON-LD specification [\[JSON-LD\]](#).

- **facet**

Category according to which search results can be categorized. See faceted search.

- **faceted search**

Faceted results for a query correspond to an analysis of how the search results are distributed over various categories (or "facets"). For example the analysis may reveal how the results are distributed by organization. The user might then refine the query to one particular organization among those listed [\[OASIS-SR-3\]](#).

- **GeoJSON**

A geospatial data interchange format based on JavaScript Object Notation (JSON) [\[GeoJSON\]](#).

- **JSON**

A lightweight, text-based, language-independent data interchange format, based on the JavaScript programming language.

- **JSON Schema**

JSON Schema is a JSON media type for defining the structure of JSON data. JSON Schema provides a contract for what JSON data is required for a given application and how to interact with it [\[JSON Schema\]](#).

- **OpenAPI Document**

A document (or set of documents) that defines or describes an API. An OpenAPI definition uses and conforms to the OpenAPI Specification [\[OpenAPI\]](#).

- **OpenSearch**

Draft specification for web search syndication, originating from Amazon's A9 project and given a corresponding interface binding by the OASIS Search Web Services working group.

- **OSDD**

OpenSearch Description Document—An XML document available at a consistent location describing metadata for the service and providing templates for queries.

- **service interface**

Shared boundary between an automated system or human being and another automated system or human being (source: ISO 19101).

4.1. Abbreviated terms

- ADES Application Deployment and Execution Service
- APD Abstract Protocol Definition
- API Application Programming Interface
- ATS Abstract Test Suite
- CEOS Committee on Earth Observation Satellites
- CPP Cloud Processing and Portrayal
- EMS Execution Management Service
- EO Earth Observation
- EOPAD Earth Observation Process and Application Discovery
- ER Engineering Report
- HATEOAS Hypermedia As The Engine Of Application State
- HTTP HyperText Transfer Protocol
- IRI Internationalised Resource Identifier
- ISO International Organisation for Standardisation
- JSON JavaScript Object Notation
- JSON-LD JavaScript Object Notation for Linked Data
- LDP Linked Data Protocol
- OASIS Organization for the Advancement of Structured Information Standards
- OGC Open Geospatial Consortium
- OWC OGC Web Services Context
- RDF Resource Description Framework
- REST Representational State Transfer
- SRU Search/Retrieval via URL
- SWG Standards Working Group
- UML Unified Modeling Language
- URI Uniform Resource Identifier
- URL Uniform Resource Locator

- URN Uniform Resource Name
- W3C World Wide Web Consortium
- WGISS Working Group on Information Systems and Services
- WKT Well-Known Text
- WPS Web Processing Service
- WPS-T Transactional Web Processing Service
- XML eXtensible Markup Language
- XSD XML Schema Definition Language

4.2. Symbols

4.2.1. JSON Schema diagrams

The schema diagrams included in the document show the JSON structure expressed in [JSON Schema](#). Section 5.2.1 of [\[OGC17-047\]](#) gives a brief overview of the notation used.

4.2.2. JSONPath

The data dictionary tables in the current document use the [\[JSONPath\]](#) notation. A brief overview of this notation is included in section 5.2.2 of [\[OGC17-047\]](#).

4.2.3. Unified Modeling Language

The Unified Modeling Language (UML) is used in some of the diagrams in the current document. Stereotypes from the [REST profile](#) [https://www.ibm.com/support/knowledgecenter/SS8PJ7_9.6.0/com.ibm.xtools.rest.doc/topics/r_rest_profile.html] are used as well.

4.3. Data dictionary tables

This document includes data dictionary tables with information as per sub-clause 5.5 of [\[OGC06-121r9\]](#). The following comment applies:

- Column 1 provides the JSON property name as well as the corresponding [\[JSONPath\]](#) expression.

Chapter 5. Overview

[Chapter 3](#) lists the normative and informative references used in this document.

[Chapter 4](#) defines the main terms used in the document.

Chapter 5 gives an overview of the document.

[Chapter 6](#) presents the Catalog Service specification defined in this testbed.

In [Chapter 7](#), the metadata model is defined in detail.

[Chapter 8](#) describes the software design and implementation of the software components deployed in the EOPAD task of the Testbed-15 thread.

Finally, [Chapter 9](#) presents the work related to the Arctic Discovery Catalogue, an evergreen catalogue of relevant Arctic circumpolar Web services from OGC and ESRI REST Web services that have some relevance to circumpolar science.

[Annex B](#) explains how the GeoJSON encoding can be interpreted as JSON-LD and includes the corresponding JSON-LD @context definitions.

[Annex C](#) contains the complete listing of examples illustrating the encodings defined in this document.

[Annex D](#) includes the JSON schema definitions defining the GeoJSON encoding.

[Annex E](#) includes a draft specification for OGC API - Processes, Transactional Profile OpenAPI 3.0 as was used in the EOPAD Task.

[Annex F](#) provides the revision history of this document.

Chapter 6. Catalogue Service Specification

6.1. Introduction

The discovery solution proposed in Testbed-15 comprises building blocks through which applications and related services can be exposed through a Catalogue service. The service consists of the following interfaces:

- Service Interface: Providing the call interface through which a catalogue client or another application can discover applications and services through faceted search and textual search, and then retrieve application/service metadata providing more detail.
- Service Management Interface: Providing the call interface through which a catalog client or other application can create, update and delete information about applications/services.

Each of the above interfaces is discussed in more detail in the following sections. The metadata model provides the data structure through which the application and/or service is described and presented as a resource in the catalog. This is defined in [chapter 7](#).

6.2. Service Interface

Testbed-15 has selected the OpenSearch interface with [\[GeoJSON\]](#) response that supports considerable reuse of existing OGC Standards. This choice avoids the creation of yet another catalog specification.

6.2.1. OpenSearch Description Document

The call interface is defined in the OpenSearch Description Document as defined in [OGC 13-026r9](#) and [OGC 10-032r8](#). Testbed-15 uses the URL template corresponding to the [application/geo+json](#) response.

The following request returns the OpenSearch Description Document representation.

```
GET https://fedeo.esa.int/api  
Accept: application/opensearchdescription+xml
```

OSDD Example

```
<?xml version="1.0" encoding="UTF-8"?><OpenSearchDescription xmlns="http://a9.com/-  
/spec/opensearch/1.1/" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:eo=  
"http://a9.com/-/opensearch/extensions/eo/1.0/" xmlns:geo="http://a9.com/-  
/opensearch/extensions/geo/1.0/" xmlns:os="http://a9.com/-/spec/opensearch/1.1/"  
xmlns:param="http://a9.com/-/spec/opensearch/extensions/parameters/1.0/"  
xmlns:semantic="http://a9.com/-/opensearch/extensions/semantic/1.0/" xmlns:sru=  
"http://a9.com/-/opensearch/extensions/sru/2.0/" xmlns:time="http://a9.com/-  
/opensearch/extensions/time/1.0/">  
<ShortName>FEDEO</ShortName>  
<Description>Provides interoperable access, following ISO/OGC interface guidelines, to
```

```

Earth Observation metadata.</Description>
<Tags>FEDEO, ESA, Earth Observation, Digital Repository, HMA, HMA-S, HMA-SE, CEOS-OS-BP-V1.1/L1.</Tags>

<Url rel="self" template="http://fedeo.esa.int/api"
type="application/opensearchdescription+xml"/>

<Url indexOffset="1" pageOffset="1" rel="results"
template="http://fedeo.esa.int/services?subject={eo:keyword?}&query={searchTerms?}&
&startRecord={startIndex?}&startPage={startPage?}&maximumRecords={count?}&
&startDate={time:start?}&endDate={time:end?}&title={eo:title?}&publishe
r={dc:publisher?}&bbox={geo:box?}&uid={geo:uid?}&organisationName={eo:orga
nisationName?}&productType={eo:productType?}&platform={eo:platform?}&instr
ument={eo:instrument?}&classifiedAs={semantic:classifiedAs?}&facetLimit={sr:facetLimit?}"
type="application/geo+json">

<param:Parameter name="platform" pattern="[a-z|A-Z|1-9|\-|_]+" title="Search by name
of the satellite or platform." value="{eo:platform}"/>
<param:Option label="ENVISAT (58)" value="ENVISAT"/>
<param:Option label="ERS-1 (10)" value="ERS-1"/>
<param:Option label="ERS-2 (22)" value="ERS-2"/>
<param:Option label="SENTINEL-1A (6)" value="SENTINEL-1A"/>
<param:Option label="SENTINEL-1B (4)" value="SENTINEL-1B"/>
</param:Parameter>

<param:Parameter name="organisationName" pattern="[a-z|A-Z|1-9|\-|_]+" title="Search
by name of the organization providing the service." value="{eo:organisationName}"/>
<param:Option label="DLR (15)" value="DLR"/>
<param:Option label="ESA/ESRIN (26)" value="ESA/ESRIN"/>
<param:Option label="VITO (30)" value="VITO"/>
</param:Parameter>

</Url>

<Query eo:platform="Envisat" role="example"/>
<LongName>Earth Observation Catalogue</LongName>

<OutputEncoding>UTF-8</OutputEncoding>
<InputEncoding>UTF-8</InputEncoding>
</OpenSearchDescription>

```

TIP The `rel` attribute of the `Url` element is set to `results`. Future versions of [OGC13-026r9] could extend the list in section 7.1 (`collection`, `results`) with a `services` value to allow clients to distinguish the URL template to be used for services/applications.

Any combination of search parameters from the [OpenSearch] specification and the following OpenSearch extensions can be used by implementations:

- [OpenSearch]

- [\[OGC10-032r8\]](#) (Geo and time)
- [\[OGC13-026r9\]](#) (Earth Observation)
- [\[OGC13-068\]](#) (Correlated search)
- [SRU Extension of OpenSearch](#) (Faceted search)
- [Semantic Extension of OpenSearch](#)

The table below lists a number of useful search parameters originating from the above specifications. The list is non-exhaustive and any combination of parameters from the above specifications can be advertised by a catalogue server.

Table 1. Search parameters

Search Parameter	Description	Metadata mapping	Examples
os:searchTerms query={searchTerms?}	Free text search.	\$.properties.abstract \$.properties.title \$.properties.categories \$.properties.keyword \$.properties.contactPoint[*].name	query=vegetation
eo:keyword subject={eo:keyword?}	Search for matching keyword.	\$.properties.keyword \$.properties.categories	keyword=vegetation
eo:title title={eo:title?}	Search for matching title.	\$.properties.title	
geo:uid uid={geo:uid?}	Retrieve metadata for service or application via its identifier.	\$.properties.identifier	

Search Parameter	Description	Metadata mapping	Examples
sru:facetLimit facetLimit={sru:facetLimit?}	Number of counts that should be reported per facet field.	Not applicable	<ul style="list-style-type: none"> • facetLimit=0 (Do not return facets) • facetLimit=10 (Return maximum 10 counts per facet) • facetLimit=eo:organisationName (Return default number of counts for eo:organisationName facet) • facetLimit=eo:organisationName:20 (Return maximum 20 counts for eo:organisationName facet)

Testbed-15 identified the need for the following additional search parameters which could not be found in existing standards/specifications that are needed to implement the EOPAD use cases:

Table 2. Additional search parameters

Search Parameter	Description	Metadata mapping	Examples
eopad:offering	Retrieve services based on type of "offering", filter based on input or output parameters to "chain" etc.	\$.properties.offerings[*].code	
eopad:type	Retrieve resources of a specific kind. This is useful if the same catalogue contains metadata for EO collections, EO products and EO services/applications.	\$.properties.kind	

6.2.2. Facet Discovery

Although the OSDD can advertise possible values for search parameters (e.g. eo:organisationName) using the [Parameter Extension](#) as depicted below, it does not advertise which facets (categories) the server can provide faceted results in the responses for.

```
<param:Parameter name="organisationName" pattern="[a-z|A-Z|1-9|\-|_]+" title="Search by name of the organization providing the service." value="{eo:organisationName}">
<param:Option label="DLR (15)" value="DLR"/>
<param:Option label="ESA/ESRIN (26)" value="ESA/ESRIN"/>
<param:Option label="VITO (30)" value="VITO"/>
</param:Parameter>
```

The [OASIS searchRetrieve Explain specification](#) defines an XML encoding that includes a dedicated searchInfo/facets section.

Choosing between different representations of the same `/api` resource is a matter of using HTTP content negotiation.

The following request returns the OSDD representation.

```
GET https://fedeo.esa.int/api
Accept: application/opensearchdescription+xml
```

The following request returns the Search/Retrieval via URL (SRU) Explain representation.

```
GET https://fedeo.esa.int/api
Accept: application/sru+xml
```

```

<explain xmlns="http://explain.z3950.org/dtd/2.0/" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=
"http://explain.z3950.org/dtd/2.0/ http://docs.oasis-open.org/search-
ws/searchRetrieve/v1.0/os/schemas/explain.xsd">
    <serverInfo>
        <host>fedeo.esa.int</host>
        <port>80</port>
        <database>/services</database>
    </serverInfo>
    <indexInfo>
        <set identifier="http://a9.com/-/opensearch/extensions/eo/1.0/" name="eo"/>
        <index search="true">
            <title lang="en">Organisation</title>
            <map>
                <name set="eo">organisationName</name>
            </map>
            <configInfo>
                <supports type="value">VITO</supports>
                <supports type="value">EUMETSAT</supports>
                <supports type="value">ESA/ESRIN</supports>
                <supports type="value">NASA</supports>
                <supports type="value">CNES</supports>
                <supports type="value">DLR</supports>
            </configInfo>
        </index>
    </indexInfo>
    <searchInfo>
        <facets>
            <facet>
                <facetType>eo:organisationName</facetType>
                <limit>
                    <limitDefault>100</limitDefault>
                    <otherLimit>true</otherLimit>
                </limit>
                <start>
                    <startDefault>1</startDefault>
                    <otherStart>false</otherStart>
                </start>
            </facet>
        </facets>
    </searchInfo>
</explain>

```

Testbed-15 proposes a JSON encoding for the Explain document which is shown below and defined in [Annex D.3](#).

WARNING: As both the Explain document and the OpenAPI document have a JSON representation, a more specific media type is required in the HTTP Accept header (See <https://tools.ietf.org/html/rfc7231#section-5.3.2.>)

- `application/sru+xml` (Explain document in XML)
- `application/json;profile="http://explain.z3950.org/dtd/2.0/"` (Explain document in JSON)
- `text/yaml` (OpenAPI document in YAML)
- `application/openapi+json;version=3.0` (OpenAPI document in JSON)

The JSON Schema for the Explain document encoding is included as [Annex D.3](#). The following request returns the JSON representation.

```
GET https://fedeo.esa.int/api
Accept: application/json;profile="http://explain.z3950.org/dtd/2.0/"
```

```
{
  "serverInfo": {
    "host": "fedeo.esa.int",
    "port": 80,
    "database": "/services"
  },
  "indexInfo": [
    {
      "set": [
        {
          "identifier": "http://a9.com/-/opensearch/extensions/eo/1.0/",
          "name": "eo"
        }
      ],
      "index": [
        {
          "search": true,
          "map": [
            {
              "set": "eo",
              "name": "organisationName"
            }
          ],
          "configInfo": {
            "supports": [
              {
                "type": "value",
                "value": "CNES"
              },
              {
                "type": "value",
                "value": "DLR"
              },
              {
                "type": "value",
                "value": "ESA/ESRIN"
              }
            ]
          }
        }
      ]
    }
  ]
}
```

```

        {
          "type": "value",
          "value": "NASA"
        },
        {
          "type": "value",
          "value": "VITO"
        }
      ]
    }
  ]
},
"searchInfo": {
  "facets": [
    {
      "facetType": "eo:organisationName",
      "limit": {
        "limitDefault": 100,
        "otherLimit": true
      },
      "start": {
        "startDefault": 1,
        "otherStart": false
      }
    }
  ]
}

```

6.3. Service Management Interface

The OSDD can only be used to advertise the search interface. To include the service management interface supporting the creation, update and deletion of metadata records in the catalogue, a more complete interface is required.

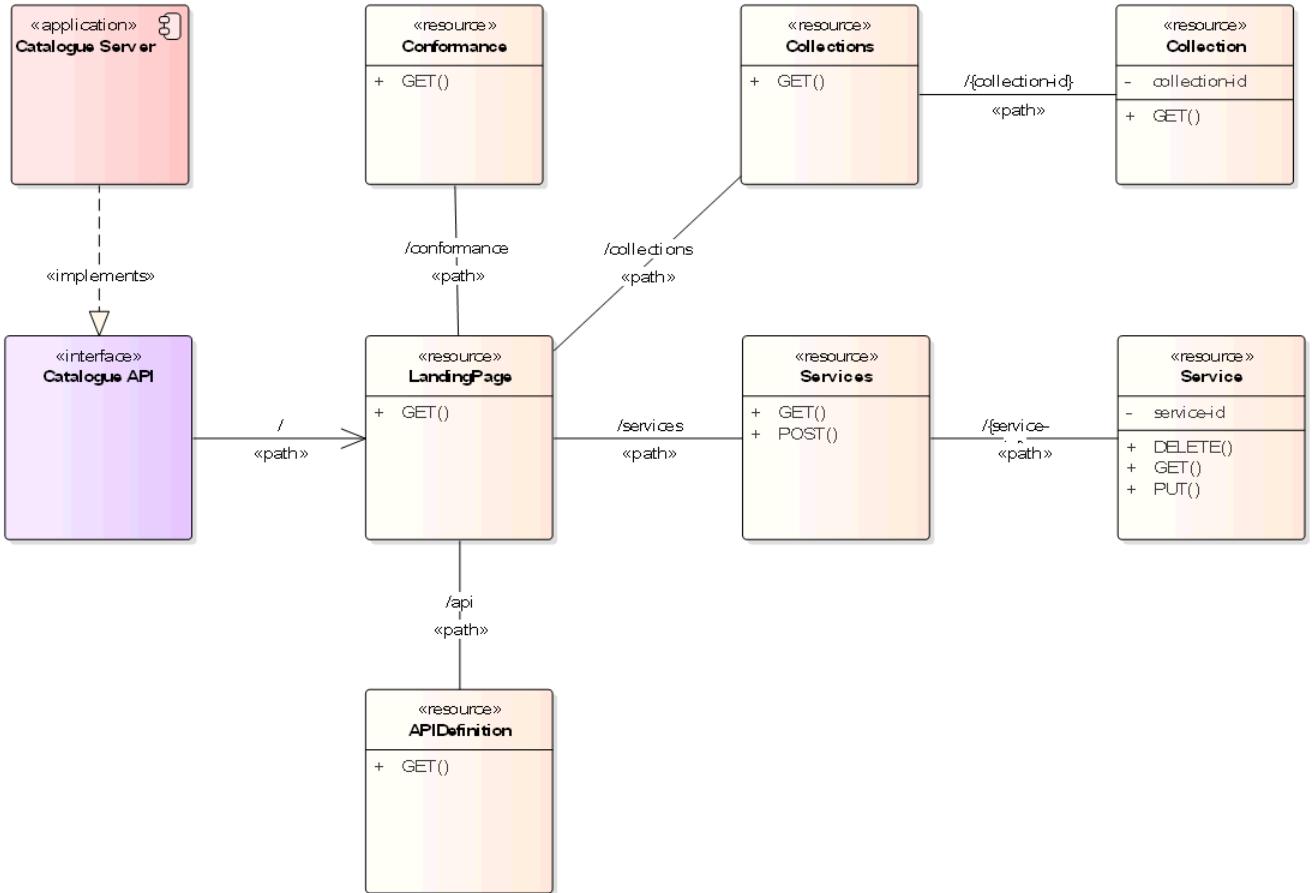


Figure 1. Resource Definition Diagram

The following paths are defined in the OpenAPI description:

```

/
/api
/conformance
/collections
/collections/{collection-id}
/services
/services/{service-id}
  
```

The following paths are included to comply with the Core requirements class of the [OGC API - Common draft specification](#). The paths are discovered via the Landing Page.

```

/
/api
/conformance
/collections
  
```

An example of a complete OpenAPI description implementing the Service Management Interface is available in [Annex C](#). The <https://editor.swagger.io/> tool can be used for visualising the OpenAPI file.

LandingPage

GET / provide landing page (rel="self") in OAPI Common format.

Conformance

GET /conformance provide conformance class information (rel="conformance") in OAPI Common format.

Collections

GET /collections Metadata about the resource collections

GET /collections/{collection-id} Metadata about a resource collection

APIDescription

GET /api describe the catalogue interface (rel="service")

Services

GET /services service or application search (rel="data")

POST /services

Service

GET /services/{service-id}

PUT /services/{service-id}

DELETE /services/{service-id}

Figure 2. OpenAPI description visualisation

6.3.1. Landing Page

The / (Landing Page) path is required to comply with the [OGC API-Common draft specification](#).

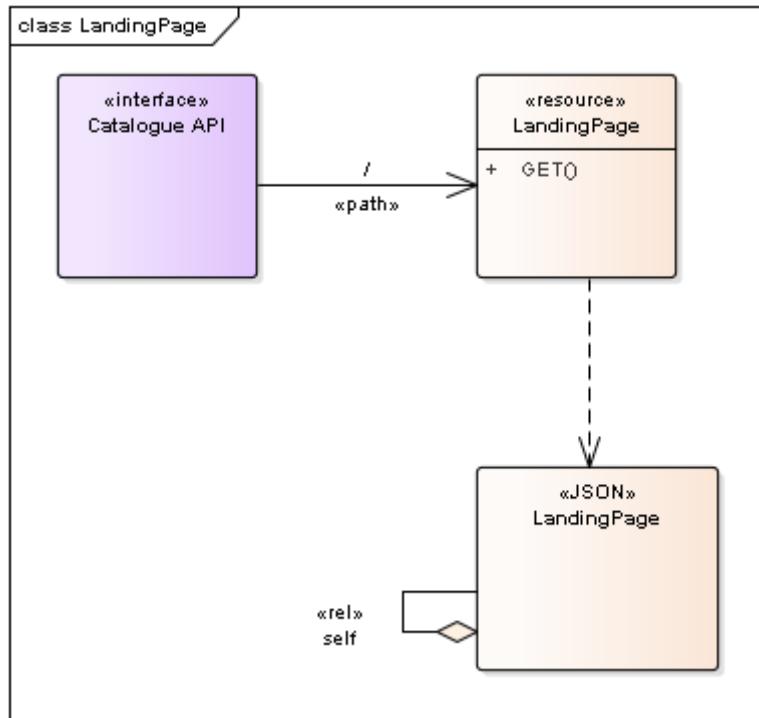


Figure 3. LandingPage Resource Specification Diagram

The following request returns the Landing Page.

```

GET https://fedeo.esa.int/
Accept: application/json

```

```
{  
  "links": [  
    { "href": "http://fedeo.esa.int/",  
      "rel": "self", "type": "application/json", "title": "this document" },  
    { "href": "http://fedeo.esa.int/api",  
      "rel": "service", "type": "application/openapi+json;version=3.0", "title":  
        "OpenAPI definition" },  
    { "href": "http://fedeo.esa.int/api",  
      "rel": "service", "type": "text/yaml", "title": "the OpenAPI definition" },  
    { "href": "http://fedeo.esa.int/api",  
      "rel": "service", "type": "application/opensearchdescription+xml", "title":  
        "OpenSearch Description Document" },  
    { "href": "http://fedeo.esa.int/api",  
      "rel": "service", "type": "application/sru+xml", "title": "Explain Document" },  
    { "href": "http://fedeo.esa.int/api",  
      "rel": "service", "type": "application/json;profile=  
\\\"http://explain.z3950.org/dtd/2.0/\\\"", "title": "Explain Document" },  
    { "href": "http://fedeo.esa.int/conformance",  
      "rel": "conformance", "type": "application/json", "title": "OGC conformance  
classes implemented by this API" },  
    { "href": "http://fedeo.esa.int/collections",  
      "rel": "data", "type": "application/json", "title": "Collection metadata" }  
  ]  
}
```

6.3.2. API definition

WARNING: The latest version of [OGC17-069r2] defines "rel": "service-desc" (and "rel": "service-doc") instead of "rel": "service" which was used in previous versions of the [OGC17-069r2] specification. Testbed-15 EOPAD implementations were not yet updated and use the original relation name "service".

The /api (API definition) path is required to comply with in development [OGC API-Common draft specification](#). Content negotiation is used to select the desired representation. The actual paths are found in the above LandingPage response document.

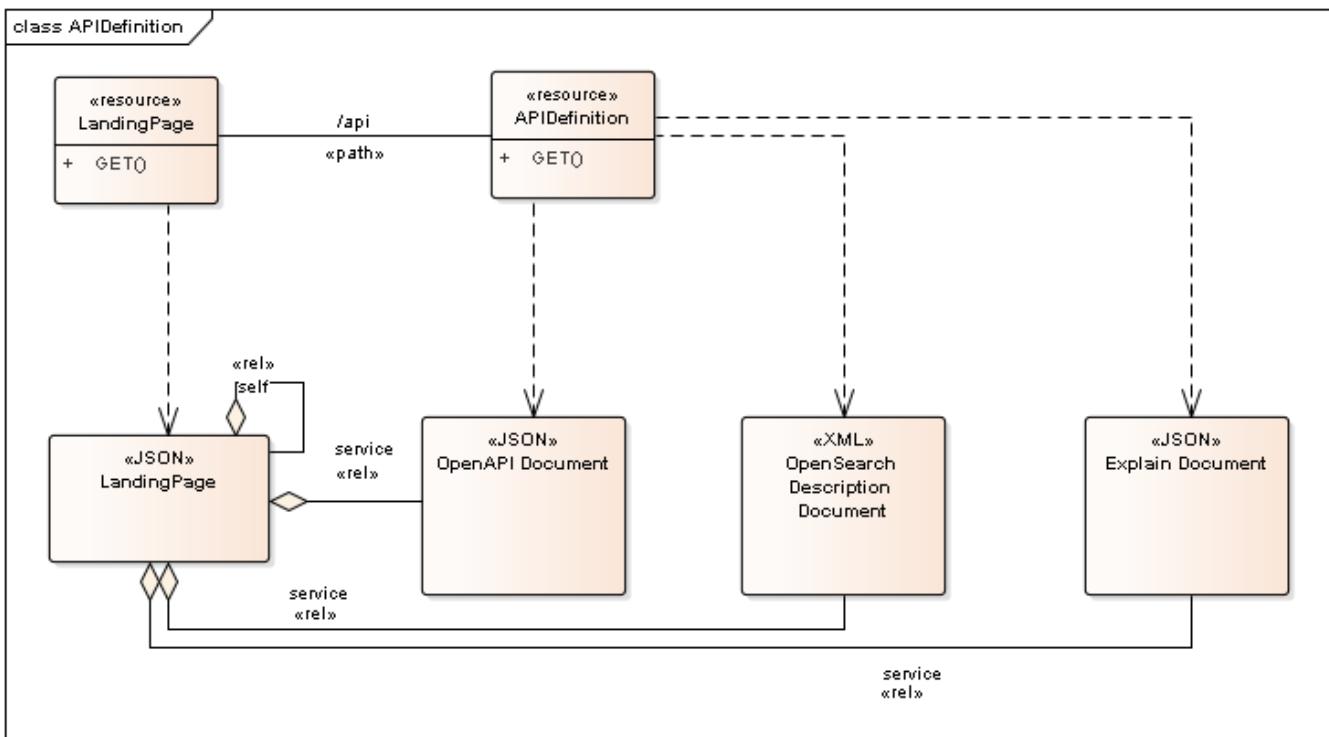


Figure 4. APIDefinition Resource Specification Diagram

The following request returns the OpenSearch Description Document representation of the APIDefinition.

```

GET https://fedeo.esa.int/api
Accept: application/opensearchdescription+xml

```

The following request returns the OpenAPI representation of the APIDefinition.

```

GET https://fedeo.esa.int/api
Accept: application/openapi+json;version=3.0

```

The following request returns the SRU Explain representation of the APIDefinition (in JSON format). The use of the **profile** parameter as part of the media type is proposed as in <https://www.w3.org/TR/json-ld/#application-ld-json>. See also [RFC-Profile].

```

GET https://fedeo.esa.int/api
Accept: application/json;profile="http://explain.z3950.org/dtd/2.0/"

```

The diagram below shows the interaction with the Landing Page and APIDefinition resources to discover information about the Catalogue API. Simple read-only catalogue clients only require the information in the OSDD document.

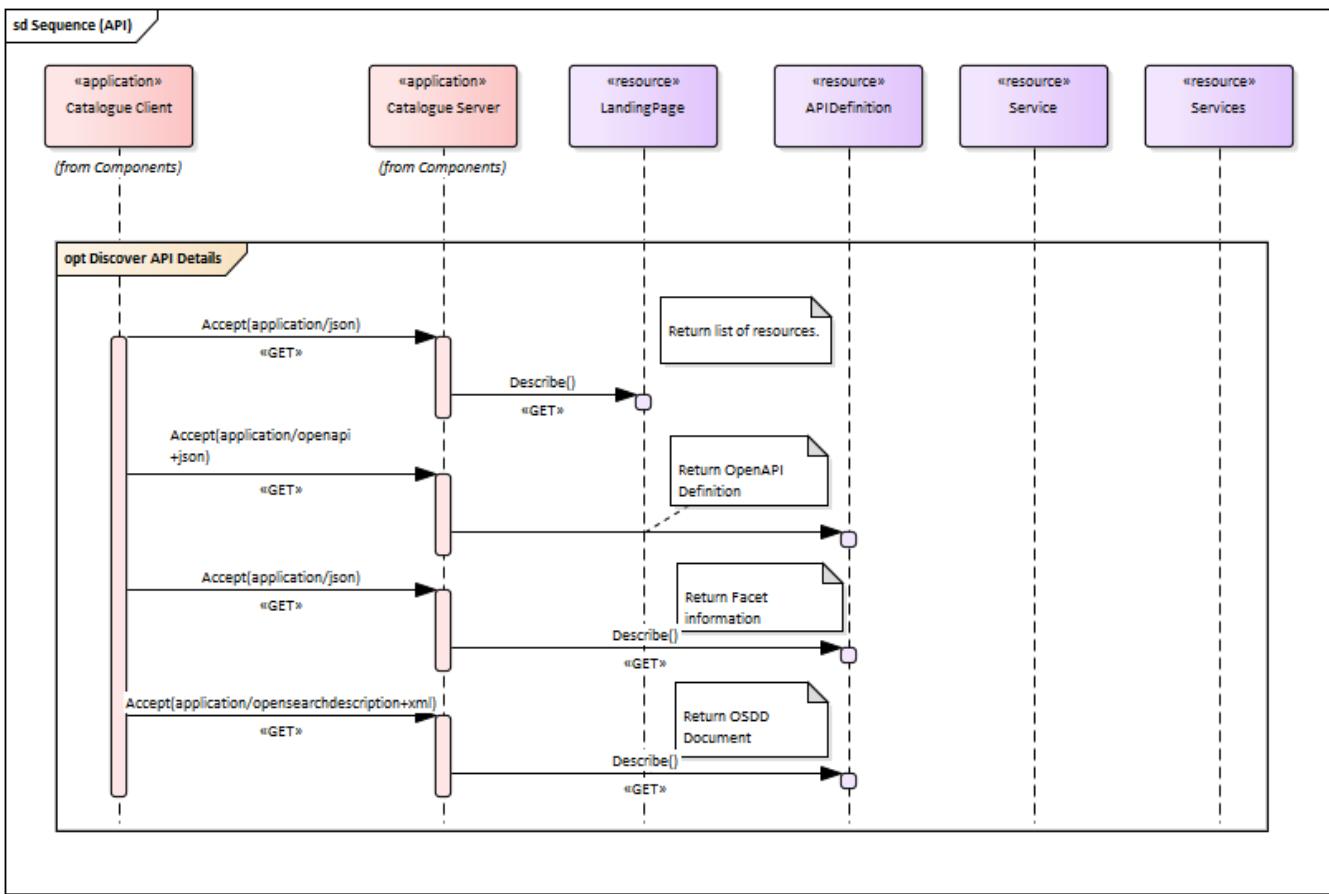


Figure 5. Sequence Diagram - Discover API

6.3.3. Conformance statements

The [/conformance](#) (Conformance statements) path is required to comply with the in development OGC API - Common draft specification.

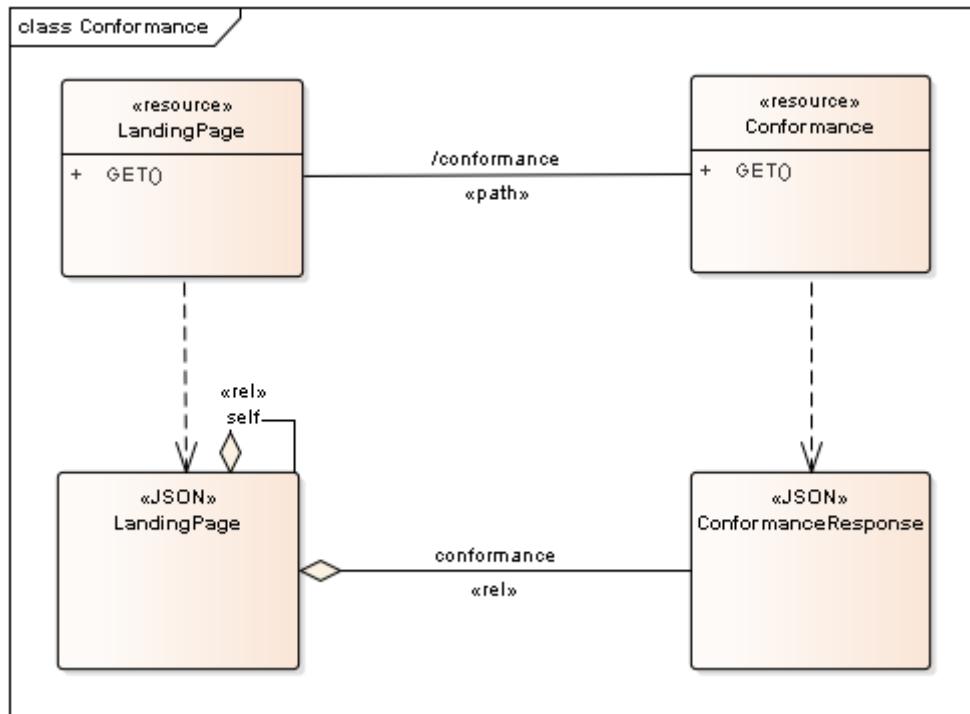


Figure 6. Conformance Resource Specification Diagram

The following request returns the Conformance statements.

```
GET https://fedeo.esa.int/conformance
```

```
Accept: application/json
```

Conformance response document

```
{  
  "conformsTo": [  
    "http://www.opengis.net/spec/owc-geojson/1.0/req/core",  
    "http://www.opengis.net/spec/os-geojson/1.0/req/core",  
    "http://www.opengis.net/spec/eopad-geojson/1.0/req/core",  
    "http://www.opengis.net/spec/OAPI_Common/1.0/req/oas30",  
    "http://www.opengis.net/spec/OAPI_Common/1.0/req/geojson"  
  ]  
}
```

6.3.4. Collections

The `/collections` path is required to comply with the in development [OGC API-Common draft specification](#). Note that these are collections as defined in [\[OAPI-Common\]](#) and not Earth Observation collections.

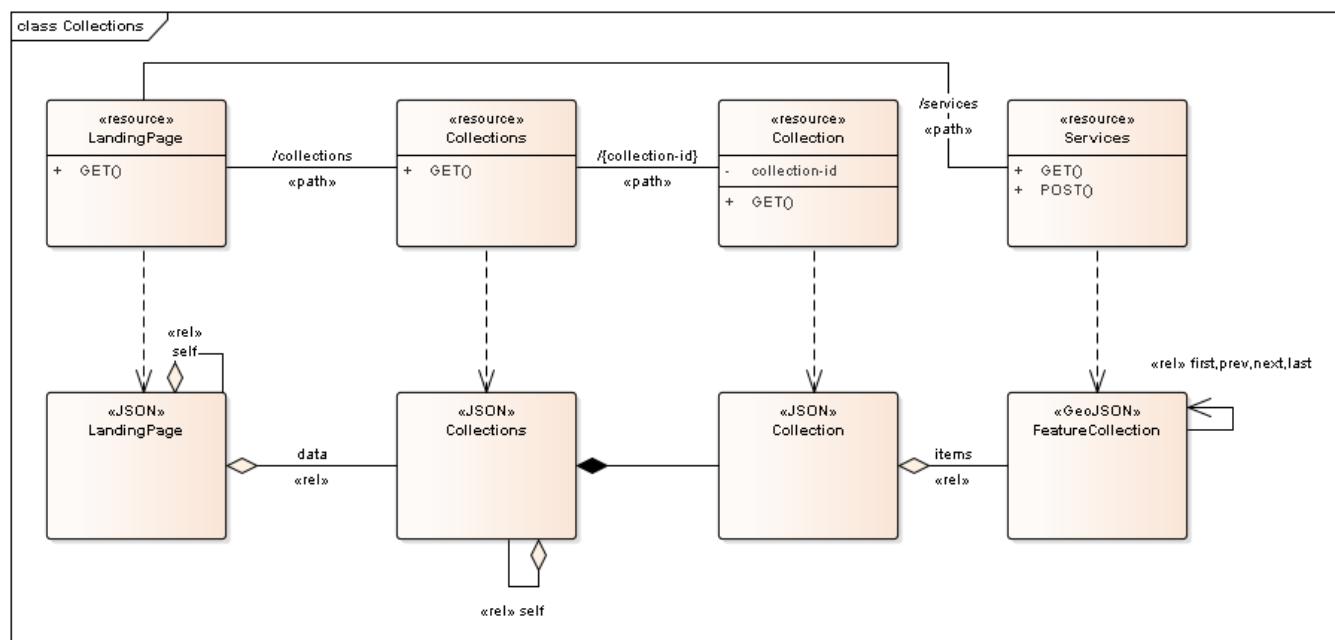


Figure 7. Collections Resource Specification Diagram

The following request returns the Collections metadata.

```
GET https://fedeo.esa.int/collections
```

```
Accept: application/json
```

```
{  
    "collections": [  
        {  
            "description": "Metadata records representing EO services and  
            applications.",  
            "links": [  
                {  
                    "rel": "items",  
                    "href": "http://fedeo.esa.int/services",  
                    "type": "application/geo+json",  
                    "title": "Services and applications"  
                },  
                {  
                    "rel": "describedBy",  
                    "href": "http://schemas.opengis.net/eopad-geojson/1.0/eopad-  
                    geojson-schema.json",  
                    "type": "application/schema+json",  
                    "title": "JSON schema for items belonging to this collection"  
                },  
                {  
                    "rel": "search",  
                    "href": "http://fedeo.esa.int/api",  
                    "type": "application/opensearchdescription+xml",  
                    "title": "Search interface"  
                }  
            ],  
            "id": "services",  
            "title": "EO services and applications"  
        }  
    ],  
    "links": [  
        {  
            "rel": "self",  
            "href": "http://fedeo.esa.int/collections",  
            "type": "application/json",  
            "title": "Collection metadata according to OGC API Common conventions"  
        }  
    ]  
}
```

6.3.5. Services

The `/services` path corresponds to a (filtered) list of services and application metadata. This path is discovered by the client in the `<Url>` element of the OSDD document or in the OpenAPI description (via the Landing Page) depending on the complexity of the Catalogue client. Content negotiation is used to select the desired representation. The `/services` path can also be discovered via the `rel="items"` link in the corresponding collection in the above Collections response document.

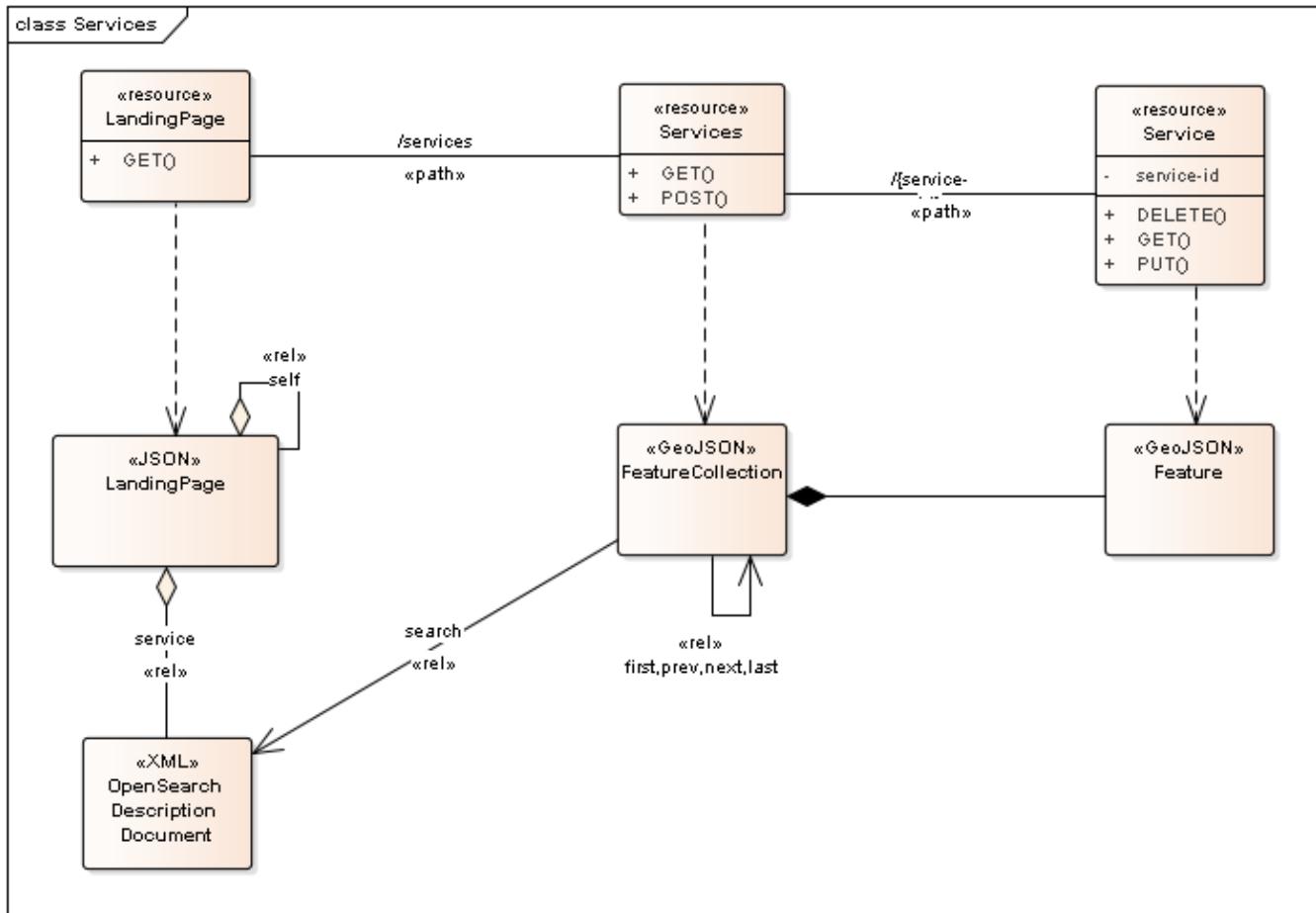


Figure 8. Services Resource Specification Diagram

The following diagram shows the interaction with the above resources to search, create, update or delete metadata items in the catalogue.

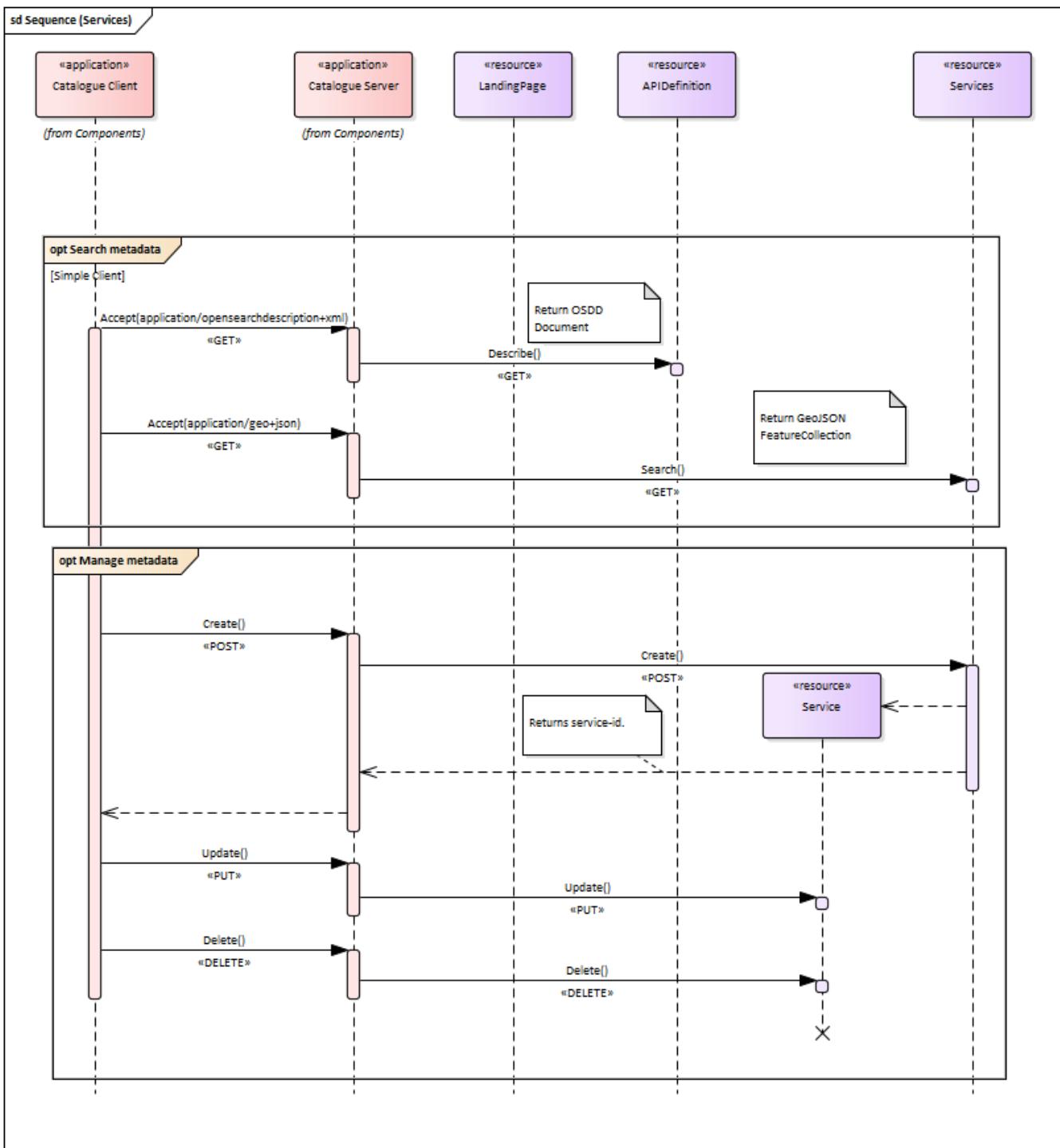


Figure 9. Sequence Diagram - Metadata discovery and management

The description of the `GET` operation on `/services` should match with the `<Url>` element information provided in the OSDD. The `x-value` attribute (in combination with `x-@context`) makes the relation with the corresponding OpenSearch search parameter from the OSDD explicit in the description. It allows defining the semantics of the query parameters appearing in the OpenAPI document by linking them to their definition in the OGC OpenSearch Standards. An example is shown below.

```
{  
  "name": "organisationName",  
  "in": "query",  
  "x-value": "{eo:organisationName}",  
  "description": "Name of data provider {eo:organisationName}.",  
  "required": false,  
  "schema": {  
    "type": "string",  
    "enum": [  
      "ESA",  
      "VITO",  
      "JAXA",  
      "DLR"  
    ]  
  }  
}
```

The following request returns a list of services or applications corresponding to the result of a free text search.

```
GET https://fedeo.esa.int/services?query=vegetation  
Accept: application/geo+json
```

The response returned is a [\[GeoJSON\]](#) FeatureCollection object and is defined in the [Data Model section 7.2](#).

6.3.6. Brief, summary responses

The OpenSearch interface with JSON response does not have the equivalent of the Catalogue Service for the Web (CSW) csw:ElementSetName parameter (brief, summary [default], full) to adjust verbosity of metadata record responses. This is typically less necessary as the JSON metadata encoding is less verbose than the XML encoding.

However, catalogue servers may decide to return less optional Feature properties and exploit the \$.properties.links.alternates or \$.properties.links.via properties to include references to alternative or original metadata in ISO or other metadata formats.

```
{
  "links": {
    "alternates": [
      {
        "href": "http://fedeo.esa.int/services/MultiSensorNDVI?httpAccept=application/vnd.iso.19139-2%2Bxml",
        "type": "application/vnd.iso.19139-2+xml",
        "title": "ISO 19139-2 metadata"
      }
    ]
  }
}
```

In case the GeoJSON response becomes large, the sru:recordSchema parameter (See [OGC13-026r9]) could be used to reduce/extend the content of the Feature object.

OpenSearch clients might also request to receive the response in Atom format which may contain an atom:link allowing to retrieve EOPAD GeoJSON metadata records in a separate request. Testbed-15 does recommend use of the GeoJSON response instead.

```
<entry>
<id>http://geo.spacebel.be/services/MultiSensorNDVI</id>
<title>Multi Sensor NDVI</title>
<summary type="html"></summary>
<content type="html">NDVI is calculated after the two bands values Near Infrared and red. It is calculated by this formula : NDVI = (NIR-Red)/(NIR+Red).</content>
<updated>2019-05-21T12:09:39Z</updated>
<dc:identifier>MultiSensorNDVI</dc:identifier>
<dc:date>2019-05-21T12:09:39Z</dc:date>
<georss:polygon>-90 -180 -90 180 90 180 90 -180 -90 -180</georss:polygon>

<link href="http://geo.spacebel.be/services/MultiSensorNDVI" rel="alternate" title="ISO 19139-2 metadata" type="application/vnd.iso.19139-2+xml"/>
<link href="http://geo.spacebel.be/services/MultiSensorNDVI" rel="alternate" title="EOPAD metadata" type="application/geo+json;profile=http://www.opengis.net/spec/eopad-geojson/1.0"/>

</entry>
```

Chapter 7. Data Model

This chapter defines the GeoJSON encoding of the different parts of the Data Model. The EOPAD metadata is defined as a JSON-LD compaction through a normative context. Therefore, the JSON-LD encoding can also be applied to other Resource Description Framework [RDF] encodings including [RDF/XML](#) and [RDF Turtle](#).

The data model makes no assumptions as to the "service" interfaces through which the metadata are accessed and applies equally well to a Service Oriented Architecture as well as a Resource Oriented or RESTful Architecture. The documented approach can be applied in combination with the following technologies:

- OGC OpenSearch [\[OGC13-026\]](#), [\[OGC17-047\]](#)
- [OpenAPI](#)
- OGC API Features [\[OGC17-069r2\]](#)
- [W3C Linked Data Platform](#)
- OASIS searchRetrieve
- OGC (Geo)SPARQL
- Discovery as Structured Data by mass market search engines
- Metadata serialised as a [label in a Dockerfile](#) [<https://docs.docker.com/config/labels-custom-metadata/>]

A presentation is used which is similar to the GeoJSON Encoding Specifications in [\[OGC17-003\]](#), [\[OGC17-047\]](#) and [\[OGC17-084\]](#). The JSON Schema encoding is included in [Annex D.1](#).

7.1. GeoJSON EOPAD Metadata Encoding

7.1.1. Design Approach

The EOPAD metadata encoding defined in this chapter consists of a Feature-based GeoJSON model. The implementation is derived from the conceptual models defined in [\[OGC11-035r1\]](#), itself based on ISO19115:2003, ISO19139:2007 and ISO19139-2. The model maximizes reuse of pre-existing standardized property names. Whenever possible, existing JSON properties from [\[GeoJSON\]](#) and OWS Context [\[OGC14-055r2\]](#) are used for modelling properties instead of new EO-specific property names. If additional properties are required, they are borrowed from [\[DCAT\]](#), [\[DCAT-AP\]](#), [\[GeoDCAT-AP\]](#), [\[DCMI\]](#), [\[PROV-O\]](#) and schema.org (in that order) with their namespace omitted while the namespace appears in the expanded JSON-LD representation. Useful types and properties from [DCAT Version 2](#), in particular related to [dcat DataService](#), are already used in combination with [\[DCAT-AP\]](#), [\[GeoDCAT-AP\]](#) even though these specifications are based on the original version of [\[DCAT\]](#).

7.1.2. Requirements class: Feature

The Feature object represents the EO application/service metadata a.k.a. EOPAD metadata. The object inherits all properties of the GeoJSON Feature object. In addition, it may contain an optional @context property. The @context properties shall typically be absent in the GeoJSON encoding and

implicitly refer to the normative @context defined in Annex B.

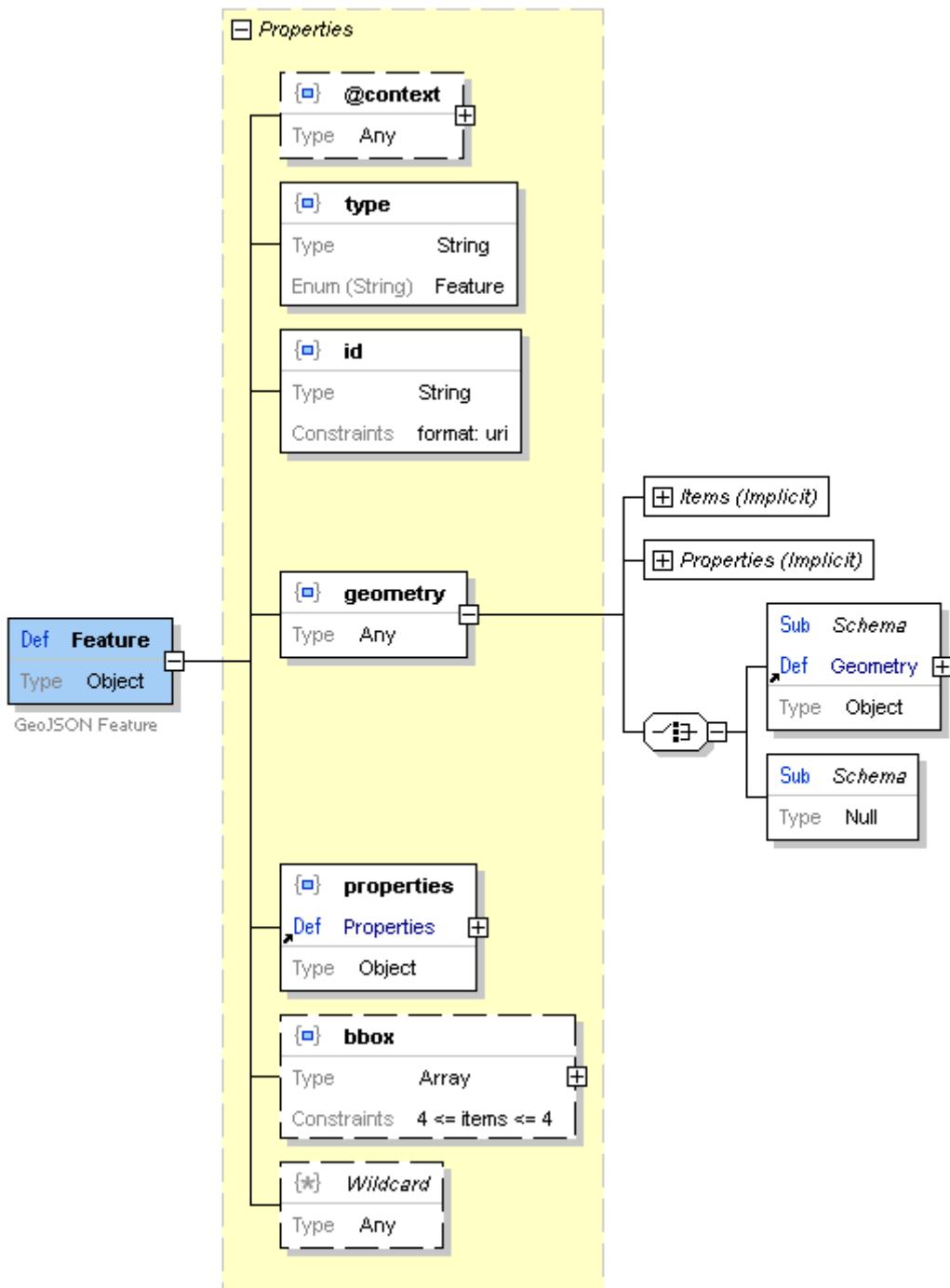


Figure 10. Feature Schema

Service/applications can be distinguished from EO collections and EO products via their Feature type. The feature type corresponds to the *dct:type* property (JSON-LD) as per the resource type and spatial data service type defined in [GeoDCAT-AP] and its JSON equivalent *kind* that is defined in OGC 17-047. A code list for services/application similar to INSPIRE Classification of Spatial Services [<http://inspire.ec.europa.eu/metadata-codelist/SpatialDataServiceCategory>] allows distinguishing different service categories, including *human interaction services*.

Table 3. Resource types

Resource	\$.type	JSON-LD @type	\$.properties.kind or JSON-LD dct:type
EO Service/Application	Feature	dcat:DataService	<ul style="list-style-type: none"> http://purl.org/dc/dcmitype/Service http://inspire.ec.europa.eu/metadata-codelist/ResourceType/service http://inspire.ec.europa.eu/metadata-codelist/SpatialDataServiceType/view
EO Collection (OGC 17-084)	Feature	dcat:Dataset	<ul style="list-style-type: none"> http://purl.org/dc/dcmitype/Collection http://inspire.ec.europa.eu/metadata-codelist/ResourceType/series
EO Product (OGC 17-003)	Feature	gj:Feature eop:EarthObservation	<ul style="list-style-type: none"> http://purl.org/dc/dcmitype/Dataset http://inspire.ec.europa.eu/metadata-codelist/ResourceType/dataset

The complete description of Feature is given in [Table 4](#). Most properties are inherited from the Feature object defined in [\[OGC14-055r2\]](#).

Table 4. Feature object properties

JSON Property	Definition	Data type and values	Multiplicity and use
@context \$.@context	Optional context property either embedding an actual context or a reference to the normative JSON-LD context defined in Annex B .	Property	Zero or one (optional)
type \$.type	Type of the element. This property is a string with fixed value "Feature".	Property Range: String	One (mandatory)
id \$.id	Unique identifier for the application or service (URI).	Property Domain: Feature Range: String (URI)	One (mandatory)

JSON Property	Definition	Data type and values	Multiplicity and use
bbox \$.bbox	Information on the coordinate range of the geometry object representing the footprint. The value is an array of length 4 (assuming the number of dimensions represented in the contained geometries is 2). Typically, south-west point and north-east point coordinates in the World Geodetic System 1984 (WGS84) coordinate reference system. The value defines a shape with edges that have constant longitude and latitude.	Property Domain: Feature Range: Array	Zero or one (optional)
geometry \$.geometry	Contains the description of the geometry of the feature. The value shall be either a Geometry object or a JSON null value.	Property Domain: Feature Range: Geometry or null value	One (mandatory)
properties \$.properties	Groups all other properties of the feature not covered by the properties higher in this table as imposed by [GeoJSON] .	Property Domain: Feature Range: Properties (See Table 5)	One (mandatory)

Feature encoding example

```
{
  "type": "Feature",
  "id": "http://fedeo.esa.int/services/MultiSensorNDVI",
  "bbox": [
    -180,
    -90,
    180,
    90
  ],
  "geometry": {...},
  "properties": {...}
}
```

Feature encoding example (with explicit @context property)

```
{  
    "@context": "https://www.opengis.net/eopad-geojson/1.0",  
    "type": "Feature",  
    "id": "http://fedeo.esa.int/services/MultiSensorNDVI",  
    "bbox": [  
        -180,  
        -90,  
        180,  
        90  
    ],  
    "geometry": {...},  
    "properties": {...}  
}
```

In the remainder of the document, the @context property is not included in the GeoJSON encoding in which case it is implied as explained in Annex B.2.

Properties

The Properties block contains the service or application properties and hypermedia links to related objects. It inherits all MetadataInformation ([Table 6](#)), ServiceIdentification ([Table 7](#)), DescriptiveKeywords ([Table 21](#)) and RelatedUrl ([Table 23](#)) properties.

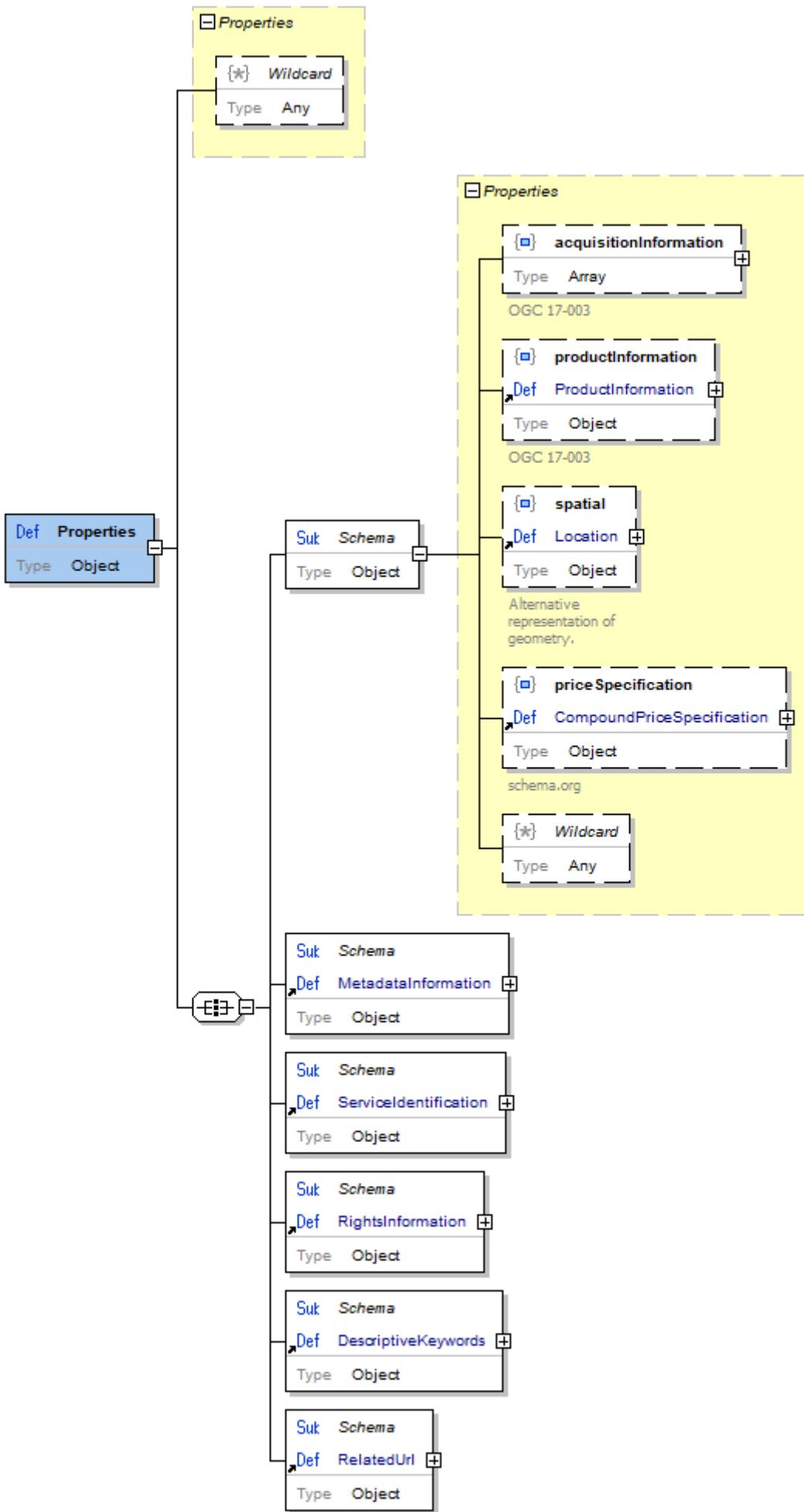


Figure 11. Properties Schema

The complete description of Properties is given in [Table 5](#).

Table 5. Properties object properties

JSON Property	Definition	Data type and values	Multiplicity and use
acquisitionInformation \$.properties.acquisitionInformation	Related acquisition information (related platform and instruments).	Property Domain: Properties Range: Array of AcquisitionInformation	Zero or one (optional)
spatial \$.properties.spatial	Alternative geometry information compliant with typical [GeoDCAT-AP] encoding.	Property Domain: Properties Range: Location	Zero or one (optional)
priceSpecification \$.properties.priceSpecification	Price information according to multiple dimensions. Defined in [schema.org] .	Range: CompoundPriceSpecification (Table 17)	Zero or one (optional)

7.1.3. Requirements class: Metadata Information

The MetadataInformation properties are inherited by the Properties block.

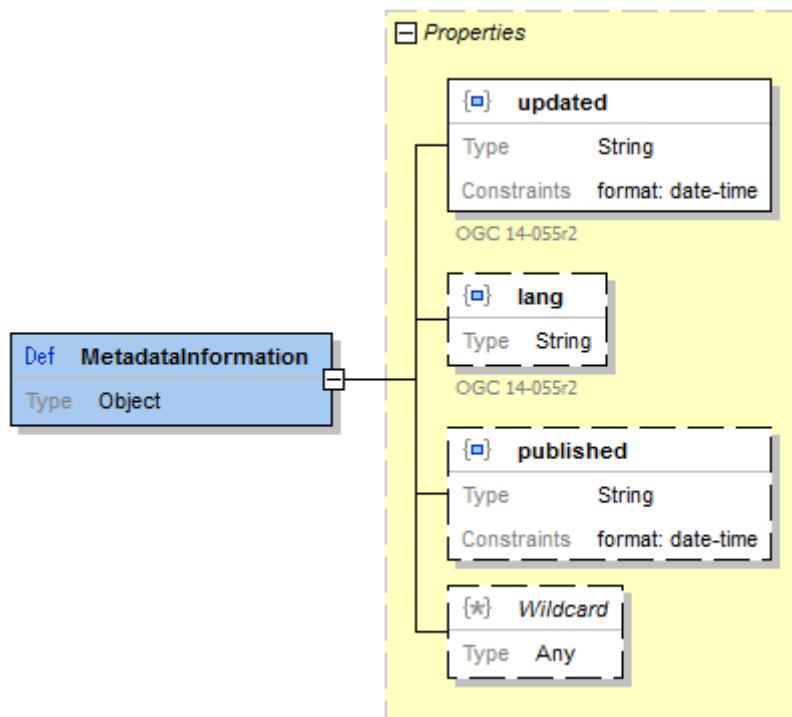


Figure 12. MetadataInformation Schema

The complete description of MetadataInformation is given in [Table 6](#).

Table 6. MetadataInformation object properties

JSON Property	Definition	Data type and values	Multiplicity and use
updated \$.properties.updated	Date of creation or last update of the metadata record. DateTime representation as defined by RFC3339 section 5.6. Property defined in [OGC14-055r2].	Range: DateTime	One (mandatory)
published \$.properties.published	Date of first availability of the metadata record.	Range: DateTime	Zero or one (optional)
lang \$.properties.lang	Metadata language, not empty with an RFC-3066 code as defined in [OGC14-055r2].	Range: String	Zero or one (optional)

MetadataInformation encoding example

```
{
  "updated": "2019-02-01T00:00:00Z",
  "lang": "en"
}
```

7.1.4. Requirements class: Service Identification

The ServiceIdentification properties are inherited by the Properties block. The ServiceIdentification inherits all properties of the ServiceContact ([Table 8](#)) object.

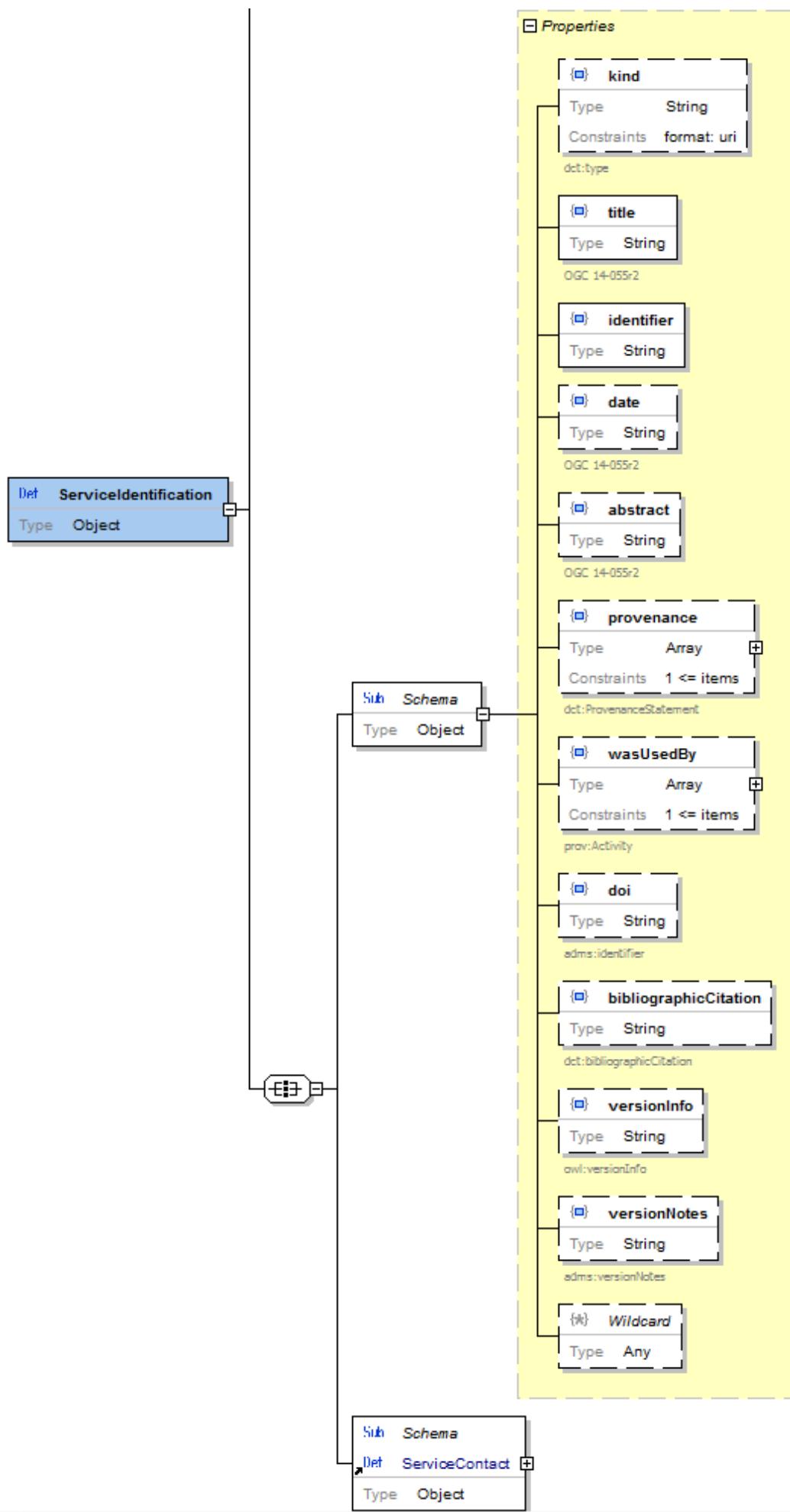


Figure 13. ServiceIdentification Schema

The complete description of ServiceIdentification is given in [Table 7](#).

Table 7. ServiceIdentification object properties

JSON Property	Definition	Data type and values	Multiplicity and use
title \$.properties.title	Human readable title given to the service or processing application. Defined in [OGC14-055r2] .	Range: String	One (mandatory)
identifier \$.properties.identifier	Identifier given to the service or processing application. Defined in [OGC14-055r2] .	Range: String	One (mandatory)
kind \$.properties.kind	Unique identifier (URI) for the type of the resource. Defined in [OGC17-047] , [DCAT-2018] and [GeoDCAT-AP] (dct:type).	Range: String	Zero or one (optional)
date \$.properties.date	Range of dates relevant for the resource (RFC-3339). Formatted as <datetime> "/"<datetime> or "<datetime>/" or "/<datetime>" or <datetime> or "/". Defined in [OGC14-055r2] .	Range: String or DateTime.	Zero or one (optional)
abstract \$.properties.abstract	Description of the service or processing application content or purpose as defined in [OGC14-055r2] .	Range: String	Zero or one (optional)
doi \$.properties.doi	Digital Object Identifier identifying the service or processing application (see https://www.doi.org). Defined in [DCAT-2018] (adms:identifier).	Range: String	Zero or one (optional)
bibliographicCitation \$.properties.bibliographicCitation	A bibliographic reference for the resource. Equivalent to mandatory ServiceCitation in [UMM-S] §2.2.1.16.	Range: String	Zero or one (optional)
versionInfo \$.properties.versionInfo	Version number or other version designation of the service or processing application. Defined in [DCAT-AP] 1.2 (owl:versionInfo).	Range: String	Zero or one (optional)
versionNotes \$.properties.versionNotes	Description of the differences between this version and a previous version of the service or processing application. Defined in [DCAT-AP] 1.2 (adms:versionNotes).	Range: String	Zero or one (optional)
provenance \$.properties.provenance	Encoding of provenance information.	Range: Array of ProvenanceStatement (Table 11)	Zero or more (optional)
wasUsedBy \$.properties.wasUsedBy	Description of the conformity of the metadata (DQ_ConformanceResult).	Range: Array of Activity (Table 12)	Zero or more (optional)

ServiceIdentification encoding example (Multi Sensor NDVI)

```
{  
  "identifier": "MultiSensorNDVI",  
  "kind": "http://inspire.ec.europa.eu/metadata-codelist/ResourceType/service",  
  "title": "Multi Sensor NDVI",  
  "abstract": "NDVI is calculated after the two bands values Near Infrared and red.  
It is calculated by this formula : NDVI = (NIR-Red)/(NIR+Red)",  
  "versionInfo": "3.0",  
  "versionNotes": "Updated for use in OGC Testbed-15."  
}
```

ServiceIdentification encoding example (Rasdaman)

```
{  
  "identifier": "59c3b4fae4b00685883826d7",  
  "kind": "http://inspire.ec.europa.eu/metadata-codelist/ResourceType/service",  
  "title": "C05.01: Rasdaman",  
  "doi": "10.5281/zenodo.1040170",  
  "bibliographicCitation": "Peter Baumann, email: p.baumann@jacobs-university.de, &  
website: rasdaman.org. (2018, January 31). rasdaman - raster data manager (Version  
9.5.0). Zenodo. http://doi.org/10.5281/zenodo.1163021",  
  "abstract": "Rasdaman (raster data manager) is an open source array database  
system, which provides flexible, fast, scalable geo services for multi-dimensional  
spatio-temporal sensor, image, simulation, and statistics data of unlimited volume.  
... data with all geo data in the PostgreSQL database, support for the raster-relevant  
OGC standards, Reference Implementation for WCS Core and WCPS\",  
}
```

ServiceContact

The ServiceContact properties are inherited by the ServiceIdentification (and Properties) block. The encoding depends on the "role" (ISO19115:2003) of the service contact. For the “publisher” and “author” roles, the OGC 14-055r2 encoding is available. Encoding for all roles is aligned with [GeoDCAT-AP] §II.16,

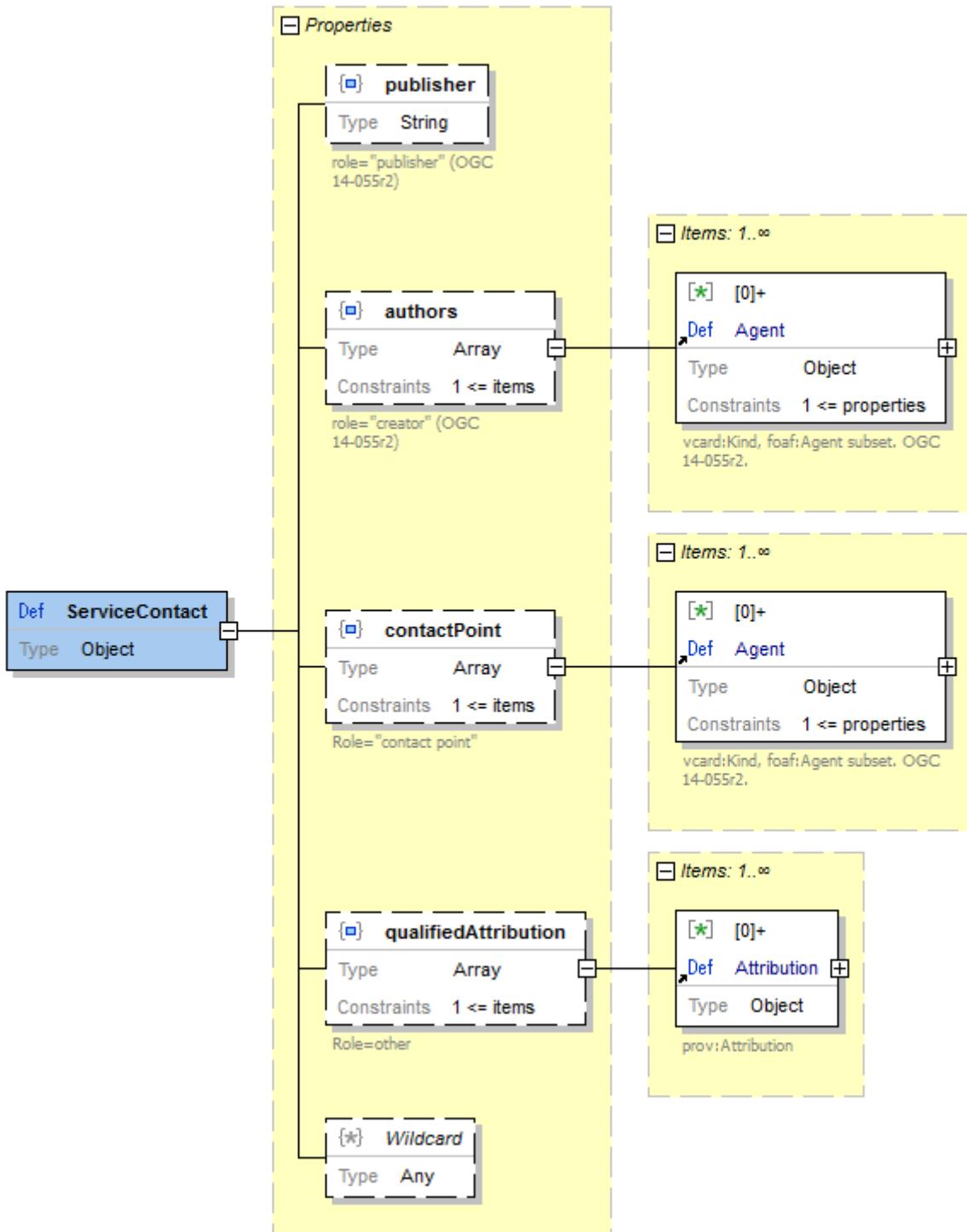


Figure 14. ServiceContact Schema

The complete description of ServiceContact is given in Table 8.

Table 8. ServiceContact object properties

JSON Property	Definition	Data type and values	Multiplicity and use
publisher \$properties.publisher	An entity or agent responsible for publishing the resource (role is "Publisher"). Defined in [OGC14-055r2].	Property Domain: Properties Range: String	Zero or one (optional)

JSON Property	Definition	Data type and values	Multiplicity and use
authors \$.properties.authors	Entities or persons primarily responsible as authors of the resource (role is "Author"). Defined in [OGC14-055r2] .	Property Domain: Properties Range: Array of Agent (Table 10)	Zero or one (optional)
contactPoint \$.properties.contactPoint	Entities or persons acting as primary contact point (role is "Point of Contact"). Defined in [DCAT] .	Property Domain: Properties Range: Array of Agent (Table 10)	Zero or one (optional)
qualifiedAttribution \$.properties.qualifiedAttribution	Specifies the relationship between the resource and the responsible organizations (role is "Resource Provider", "Custodian", "User", "Distributor", "Originator", "Principal Investigator" or "Processor"). Defined in [GeoDCAT-AP] .	Property Domain: Properties Range: Array of Attribution (Table 9)	Zero or one (optional)

ServiceContact encoding example

```
{  
  "authors": [  
    {  
      "type": "Organization",  
      "email": "mailto:eohelp@eo.esa.int",  
      "name": "ESA/ESRIN",  
      "phone": "tel:+39 06 94180777"  
    }  
  ],  
  "contactPoint": [  
    {  
      "type": "Organization",  
      "email": "mailto:eohelp@eo.esa.int",  
      "name": "ESA/ESRIN",  
      "phone": "tel:+39 06 94180777",  
      "uri": "http://www.earth.esa.int",  
      "hasAddress": {  
        "country-name": "Italy",  
        "postal-code": "00044",  
        "locality": "Frascati",  
        "street-address": "Via Galileo Galilei CP. 64"  
      }  
    }  
  ],  
  "qualifiedAttribution": [  
    {  
      "type": "Attribution",  
      "agent": [  
        {  
          "type": "Organization",  
          "email": "mailto:eohelp@eo.esa.int",  
          "name": "ESA/ESRIN",  
          "phone": "tel:+39 06 94180777",  
          "uri": "http://www.earth.esa.int",  
          "hasAddress": {  
            "country-name": "Italy",  
            "postal-code": "00044",  
            "locality": "Frascati",  
            "street-address": "Via Galileo Galilei CP. 64"  
          }  
        }  
      ],  
      "role": "originator"  
    }  
  ]  
}
```

Attribution

The Attribution object represents a responsible party and its role and is based on W3C [PROV-O].

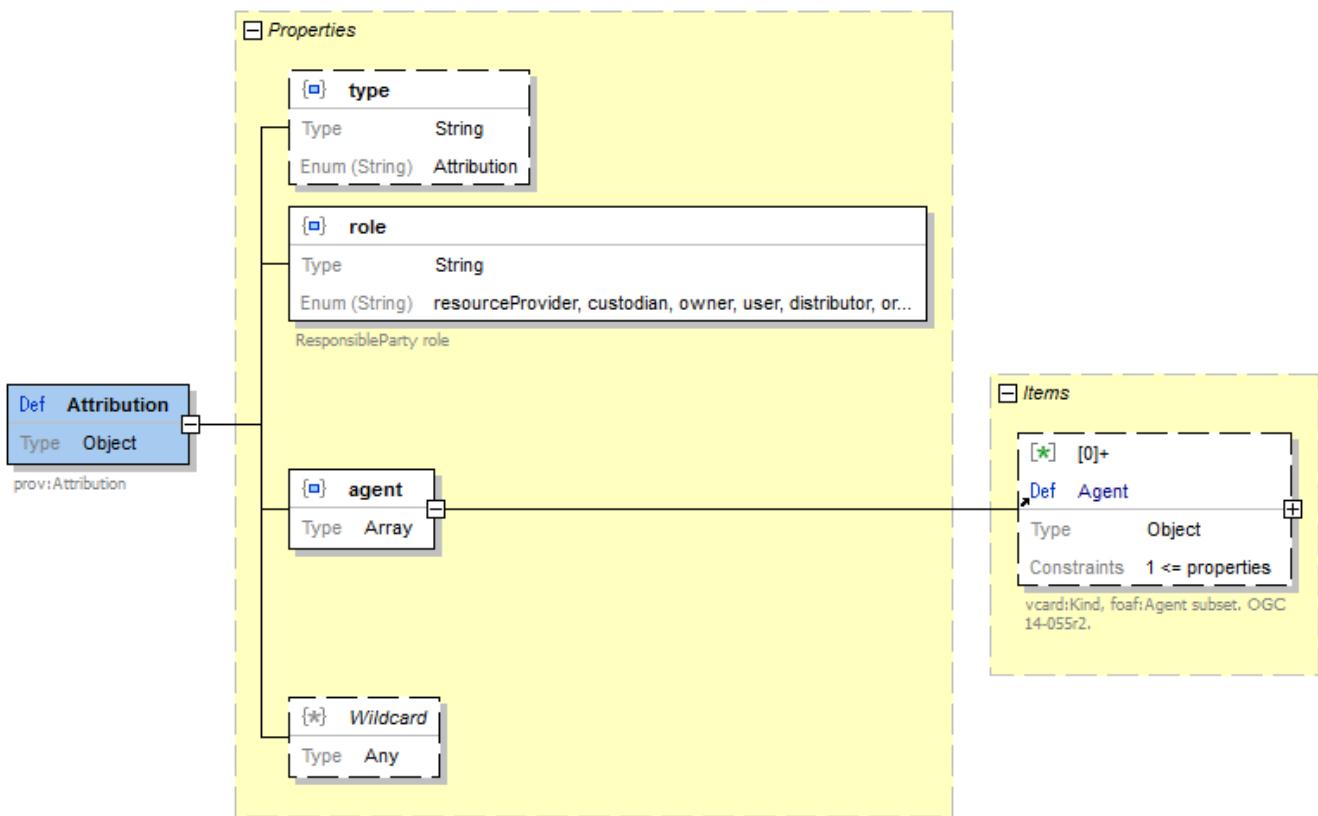


Figure 15. Attribution Schema

The complete description of Attribution is given in [Table 9](#).

Table 9. Attribution object properties

JSON Property	Definition	Data type and values	Multiplicity and use
type \$.properties.qualifiedAttribution[*].type	Type of the element. This property is a string with fixed value "Attribution".	Property Domain: Attribution Range: String Fixed value: "Attribution"	Zero or one (optional)

JSON Property	Definition	Data type and values	Multiplicity and use
role \$.properties.qualifiedAttribution[*].role	Role of the responsible party corresponding to ISO19115:2003 responsible party role. See values at See values at http://inspire.ec.europa.eu/metadata-codelist/ResponsiblePartyRole . Values: - resourceProvider - custodian - owner - user - distributor - originator - pointOfContact - principalInvestigator - processor - publisher - author	Property Domain: Attribution Range: String	One (mandatory)
agent \$.properties.qualifiedAttribution[*].agent	Organization or person fulfilling the above "role". Any properties allowed by vcard:Kind are allowed.	Property Domain: Attribution Range: Array of Agent (Table 10)	One (mandatory)

Attribution encoding example

```
{
  "type": "Attribution",
  "agent": [ {
    "type": "Organization",
    "email": "mailto:eohelp@eo.esa.int",
    "name": "ESA/ESRIN",
    "phone": "tel:+39 06 94180777",
    "uri": "http://www.earth.esa.int",
    "hasAddress": {
      "country-name": "Italy",
      "postal-code": "00044",
      "locality": "Frascati",
      "street-address": "Via Galileo Galilei CP. 64"
    }
  } ],
  "role": "originator"
}
```

Agent

The Agent object is a parent class which can contain the properties of a person or organization entity. This object is based on vcard:Kind and foaf:Agent (depending on the context) and some of its properties were defined/renamed in [\[OGC14-055r2\]](#).

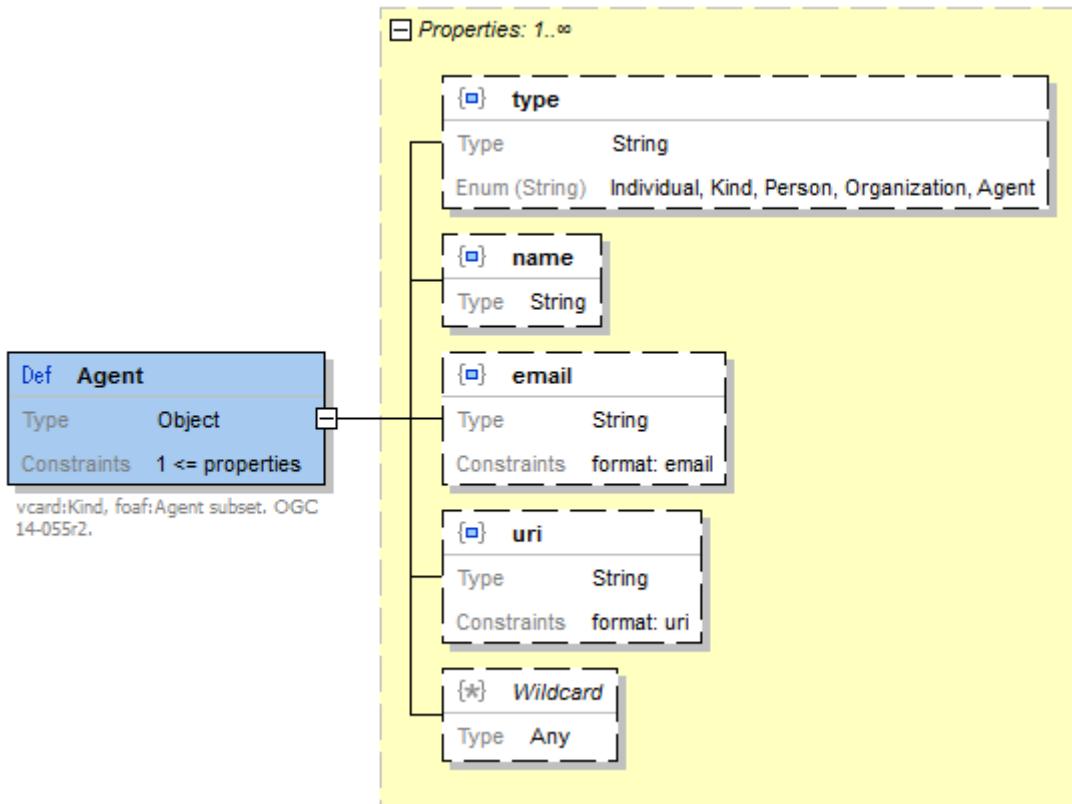


Figure 16. Agent Schema

The complete description of Agent is given in [Table 10](#).

Table 10. Agent object properties

JSON Property	Definition	Data type and values	Multiplicity and use
type \$.type	Type of the element. This property can have one of the following fixed values "Agent" (or "Kind"), "Person" (or "Individual"), "Organization".	Domain: Agent Range: String	Zero or one (optional)
name \$.name	Human readable name of author (or entity). Defined in section 7.1.1.7 of [OGC14-055r2] .	Domain: Agent Range: String	Zero or one (optional)
email \$.email	Email address of author (or entity). Defined in [OGC14-055r2] .	Domain: Agent Range: String	Zero or one (optional)
uri \$.uri	URI of author (or entity). Defined in [OGC14-055r2] .	Domain: Agent Range: String (URI)	Zero or one (optional)

Agent encoding example

```
{  
    "type": "Agent",  
    "name": "Earth Observation Helpdesk",  
    "email": "EOHelp@esa.int",  
    "uri": "http://earth.esa.int"  
}
```

ProvenanceStatement

The proposed encoding is aligned with [\[GeoDCAT-AP\]](#) §II.12 and [\[DCMI\]](#).

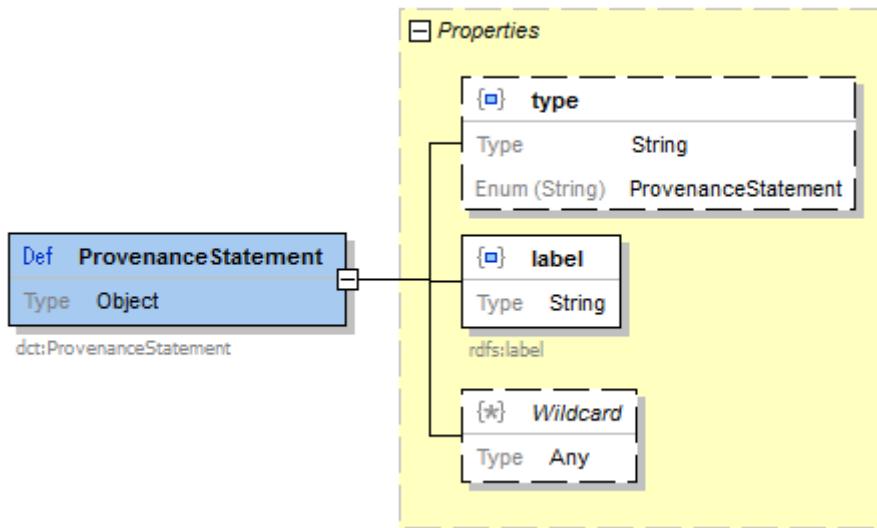


Figure 17. ProvenanceStatement Schema

The complete description of ProvenanceStatement is given in [Table 11](#).

Table 11. ProvenanceStatement object properties

JSON Property	Definition	Data type and values	Multiplicity and use
type \$.properties.provenance[*].type	Type of the element. This property has the fixed value "ProvenanceStatement".	Property Domain: ProvenanceStatement Range: String	Zero or one (optional)
label \$.properties.provenance[*].label	Free text content of the corresponding ISO19139-2 metadata property gmd:lineage (for EO collections) or [UMM-S] Service/ServiceQuality/Lineage information for EO services and applications.	Property Domain: ProvenanceStatement Range: String	One (mandatory)

ProvenanceStatement encoding example

```
{
  "type": "ProvenanceStatement",
  "label": "The software has been reviewed for quality according to the
  ISO9001:2015 compliant software verification process and no issues were reported."
}
```

Activity

The proposed encoding is aligned with [\[GeoDCAT-AP\]](#) §II.14 and [\[PROV-O\]](#).

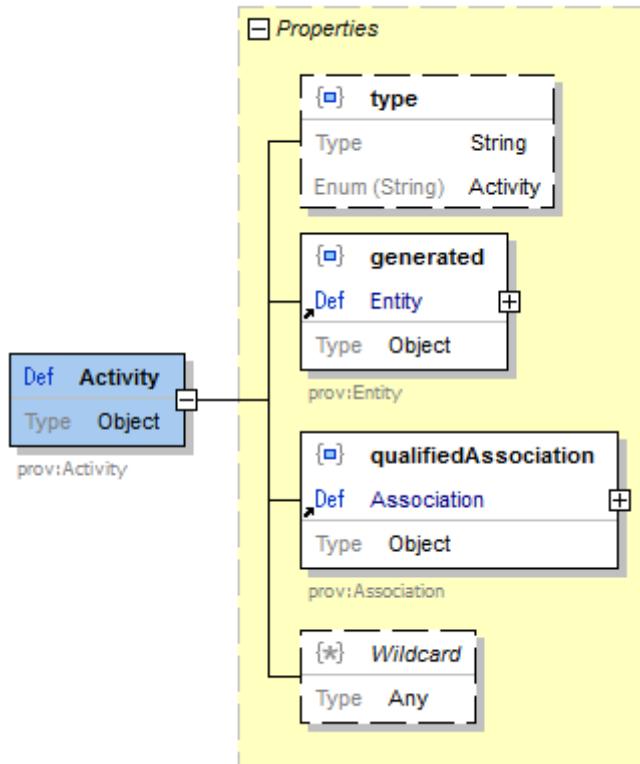


Figure 18. Activity Schema

The complete description of Activity is given in [Table 12](#).

Table 12. Activity object properties

JSON Property	Definition	Data type and values	Multiplicity and use
type \$.properties.wasUsedBy[*].type	Type of the element. This property has the fixed value "Activity".	Property Domain: Activity Range: String	Zero or one (optional)
generated \$.properties.wasUsedBy[*].generated	Refers to the conformity result which is encoded as an Entity.	Property Domain: Activity Range: Entity (Table 13)	One (mandatory)

JSON Property	Definition	Data type and values	Multiplicity and use
qualifiedAssociation \$.properties.wasUsedBy[*].qualifiedAssociation	Refers to the specification against which conformity is expressed, encoded as an Association.	Property Domain: Activity Range: Association (Table 14)	One (mandatory)

Activity encoding example

```
{
  "type": "Activity",
  "generated": {
    "type": "Entity",
    "degree": "http://inspire.ec.europa.eu/metadata-codelist/DegreeOfConformity/conformant",
    "description": "See the referenced specification"
  },
  "qualifiedAssociation": {
    "type": "Association",
    "hadPlan": {
      "type": "Plan",
      "wasDerivedFrom": {
        "type": "Standard",
        "title": "COMMISSION REGULATION (EU) No 1089/2010 of 23 November 2010 implementing Directive 2007/2/EC of the European Parliament and of the Council as regards interoperability of spatial data sets and services",
        "issued": "2010-12-08"
      }
    }
  }
}
```

Entity

The proposed encoding is aligned with [\[GeoDCAT-AP\]](#) §II.14 and [\[PROV-O\]](#).

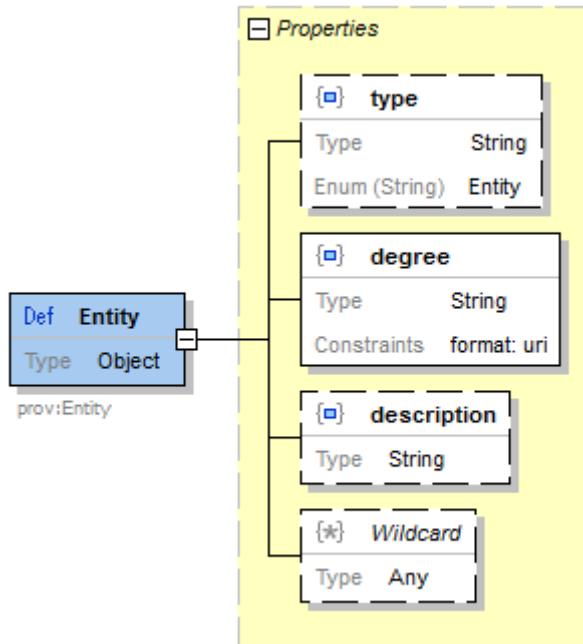


Figure 19. Entity Schema

The complete description of Entity is given in [Table 13](#).

[Table 13. Entity object properties](#)

JSON Property	Definition	Data type and values	Multiplicity and use
type \$.properties.wasUsedBy[*].generated.type	Type of the element. This property has the fixed value "Entity".	Property Domain: Entity Range: String	Zero or one (optional)
degree \$.properties.wasUsedBy[*].generated.degree	URI defining the degree of conformity. The INSPIRE Registry maintains a URI set at http://inspire.ec.europa.eu/metadata-codelist/DegreeOfConformity .	Property Domain: Entity Range: String (URI)	One (mandatory)
description \$.properties.wasUsedBy[*].generated.description	Description of the result.	Property Domain: Entity Range: String	Zero or one (optional)

Entity encoding example

```
{
  "type": "Entity",
  "degree": "http://inspire.ec.europa.eu/metadata-
  codelist/DegreeOfConformity/conformant",
  "description": "See the referenced specification"
}
```

Association

The proposed encoding is aligned with [\[GeoDCAT-AP\]](#) §II.14 and [\[PROV-O\]](#).

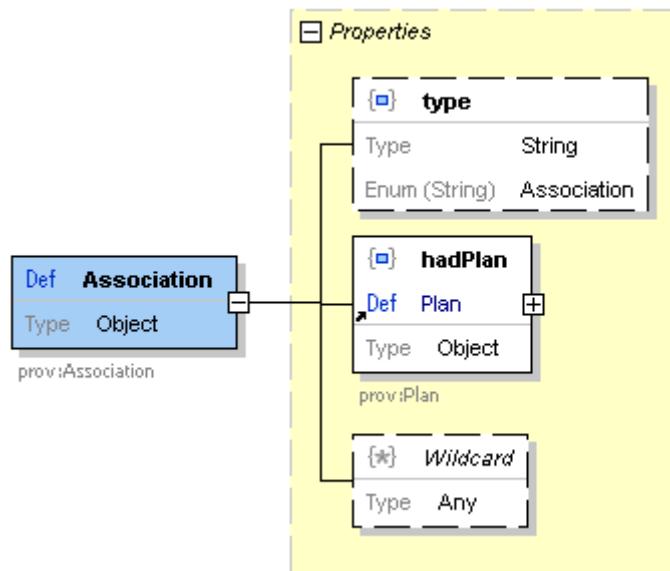


Figure 20. Association Schema

The complete description of Association is given in [Table 14](#).

Table 14. Association object properties

JSON Property	Definition	Data type and values	Multiplicity and use
type \$.properties.wasUsedBy[*].qualifiedAssociation.type	Type of the element. This property has the fixed value "Association".	Domain: Association Range: String	Zero or one (optional)
hadPlan \$.properties.wasUsedBy[*].qualifiedAssociation.hadPlan	Set of actions or steps intended by one or more agents to achieve some goals.	Domain: Association Range: Plan (Table 15)	One (mandatory)

Plan

The proposed encoding is aligned with [\[GeoDCAT-AP\]](#) §II.14 and [\[PROV-O\]](#).

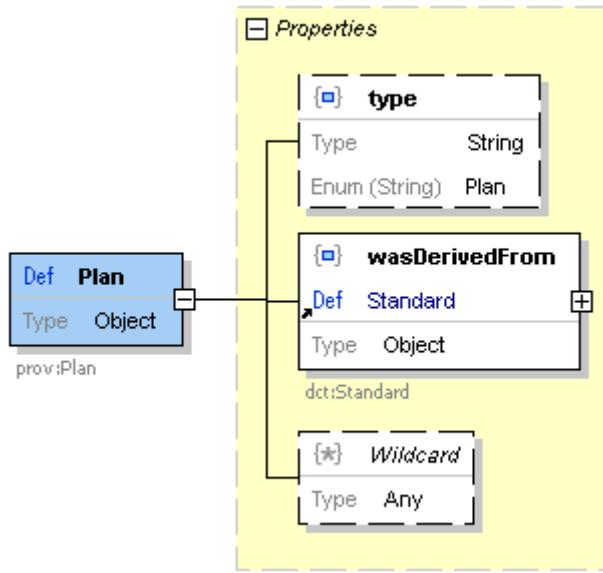


Figure 21. Plan Schema

The complete description of Plan is given in [Table 15](#).

Table 15. Plan object properties

JSON Property	Definition	Data type and values	Multiplicity and use
type \$.properties.wasUsedBy[*].qualifiedAssociation.hadPlan.type	Type of the element. This property has the fixed value "Plan".	Domain: Plan Range: String	Zero or one (optional)
wasDerivedFrom \$.properties.wasUsedBy[*].qualifiedAssociation.hadPlan.wasDerivedFrom	Refers to the specification against which conformity is expressed, encoded as a Standard.	Domain: Plan Range: Standard (Table 16)	One (mandatory)

Plan encoding example

```
{
  "type": "Plan",
  "wasDerivedFrom": {
    "type": "Standard",
    "title": "COMMISSION REGULATION (EU) No 1089/2010 of 23 November 2010
implementing Directive 2007/2/EC of the European Parliament and of the Council as
regards interoperability of spatial data sets and services",
    "issued": "2010-12-08"
  }
}
```

Standard

The proposed encoding is aligned with [\[GeoDCAT-AP\]](#) §II.14 and [\[DCMI\]](#).

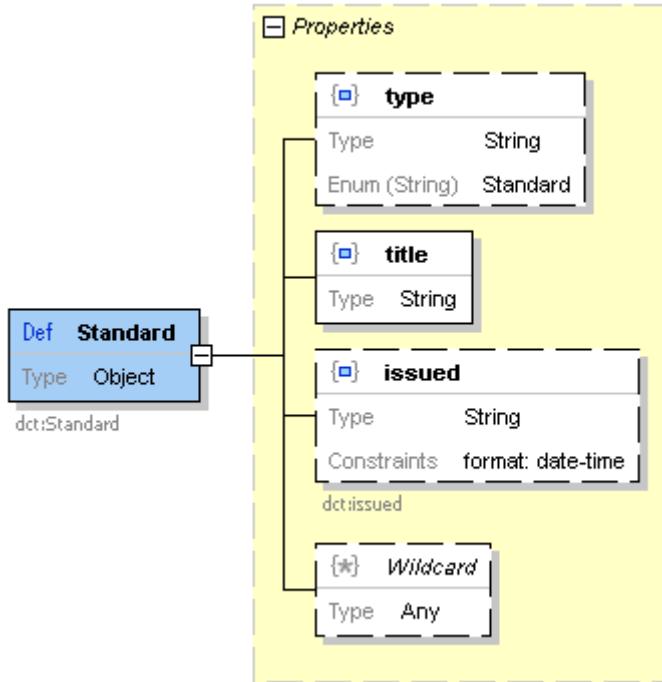


Figure 22. Standard Schema

The complete description of Standard is given in [Table 16](#).

[Table 16. Standard object properties](#)

JSON Property	Definition	Data type and values	Multiplicity and use
type \$properties.wasUsedBy[*].qualifiedAssociation.hadPlan.wasDerivedFrom.type	Type of the element. This property has the fixed value "Standard".	Domain: Standard Range: String	Zero or one (optional)
title \$properties.wasUsedBy[*].qualifiedAssociation.hadPlan.wasDerivedFrom.title	/gmd:MD_Metadata/gmd:dataQualityInfo/gmd:DQ_DataQuality/gmd:report/gmd:DQ_DomainConsistency/gmd:result Title of the specification against which conformity is expressed.	Domain: Standard Range: String	One (mandatory)
issued \$properties.wasUsedBy[*].qualifiedAssociation.hadPlan.wasDerivedFrom.issued	/gmd:MD_Metadata/gmd:dataQualityInfo/gmd:DQ_DataQuality/gmd:report/gmd:DQ_DomainConsistency/gmd:result Issued date of the specification against which conformity is expressed.	Domain: Standard Range: DateTime	Zero or one (optional)

Standard encoding example

```
{
  "type": "Standard",
  "title": "COMMISSION REGULATION (EU) No 1089/2010 of 23 November 2010
implementing Directive 2007/2/EC of the European Parliament and of the Council as
regards interoperability of spatial data sets and services",
  "issued": "2010-12-08"
}
```

CompoundPriceSpecification

The CompoundPriceSpecification element allows representing metrics for costs estimation (e.g. data size, processing cost). It groups multiple prices that all apply in combination for different dimensions of the service or application consumption. The name property of the attached unit price specification indicates the dimension of a price component. The proposed encoding is aligned with the CompoundPriceSpecification defined at <https://schema.org/CompoundPriceSpecification>.

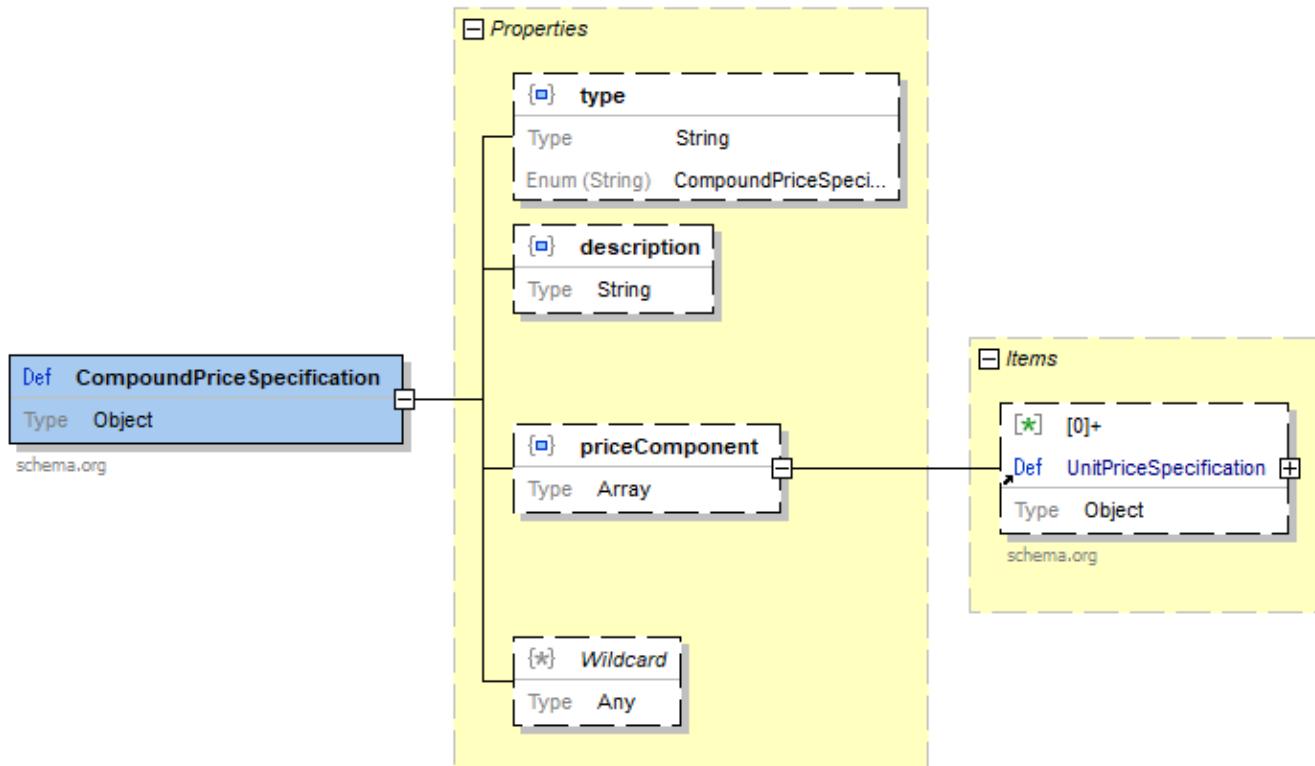


Figure 23. CompoundPriceSpecification Schema

The main properties of CompoundPriceSpecification are listed in Table 17. For additional properties, please refer to [\[schema.org\]](https://schema.org).

Table 17. CompoundPriceSpecification object properties

JSON Property	Definition	Data type and values	Multiplicity and use
type \$.properties.priceSpecification.type	Type of the element. This property has the fixed value "CompoundPriceSpecification".	Domain: CompoundPriceSpecification Range: String	Zero or one (optional)
description \$.properties.priceSpecification.description	Textual description.	Domain: CompoundPriceSpecification Range: String	Zero or one (optional)

JSON Property	Definition	Data type and values	Multiplicity and use
priceComponent \$.properties.priceSpecification.priceComponent	Price components according to multiple dimensions that apply in combination to determine the total price.	Domain: CompoundPriceSpecification Range: array of UnitPriceSpecification	Zero or one (optional)

CompoundPriceSpecification encoding example

```
{
  "type": "CompoundPriceSpecification",
  "description": "Service cost elements include processing time and input data size.",
  "priceComponent": [
    {
      "type": "UnitPriceSpecification",
      "name": "data",
      "billingIncrement": 1,
      "unitText": "km2",
      "price": 0.001,
      "priceCurrency": "EUR",
      "valueAddedTaxIncluded": true,
      "description": "Service cost element proportional to the surface represented by the processed imagery data."
    },
    {
      "type": "UnitPriceSpecification",
      "name": "processing",
      "billingIncrement": 0.1,
      "unitCode": "SEC",
      "unitText": "vCPU seconds",
      "price": 0.00002400,
      "priceCurrency": "EUR",
      "valueAddedTaxIncluded": true,
      "description": "Service cost element proportional to processing time."
    }
  ]
}
```

7.1.5. Requirements class: Rights Information

The proposed encoding is aligned with [\[DCAT-2\]](#).

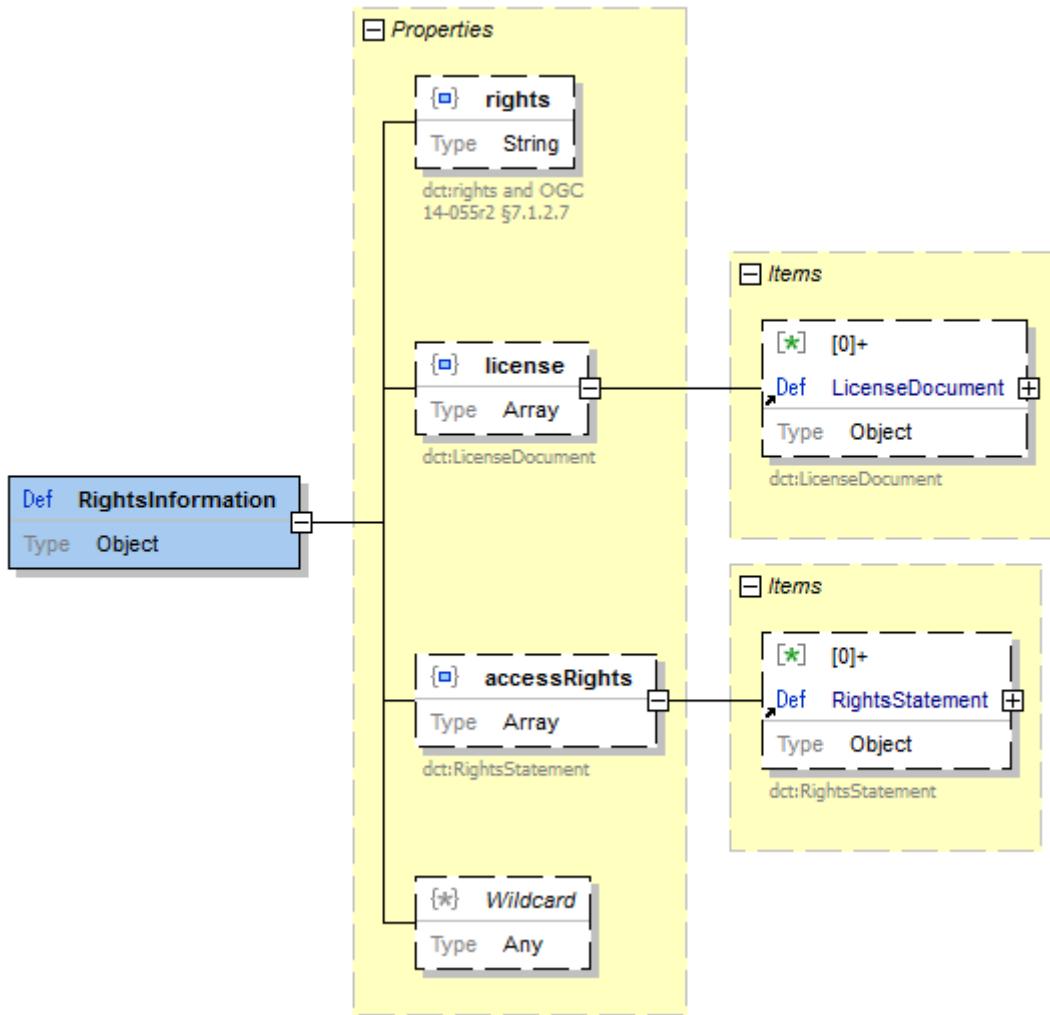


Figure 24. RightsInformation Schema

The complete description of `RightsInformation` is given in [Table 18](#).

Table 18. `RightsInformation` object properties

JSON Property	Definition	Data type and values	Multiplicity and use
<code>license</code> <code>\$properties.license</code>	Information about licensing terms for the application or service. For example the encoding of the corresponding ISO19139-2 metadata property <code>gmd:useLimitation</code> .	Domain: Properties Range: Array of LicenseDocument (Table 19)	Zero or one (optional)
<code>accessRights</code> <code>\$properties.accessRights</code>	Information about access conditions for the application or service. For example the encoding of the corresponding ISO19139-2 metadata properties <code>gmd:accessConstraints</code> and <code>gmd:otherConstraints</code> .	Domain: Properties Range: Array of RightsStatement (Table 20)	Zero or one (optional)

JSON Property	Definition	Data type and values	Multiplicity and use
rights \$.properties.rights	Information about rights held in and over the application or service. See [OGC14-055r2] §7.1.2.7. Also defined in [DCAT-2] as a statement that concerns all rights not addressed with license or accessRights, such as copyright statements.	Property Domain: Properties Range: String	Zero or one (optional)

LicenseDocument

The proposed encoding is aligned with [\[GeoDCAT-AP\]](#) §II.15.

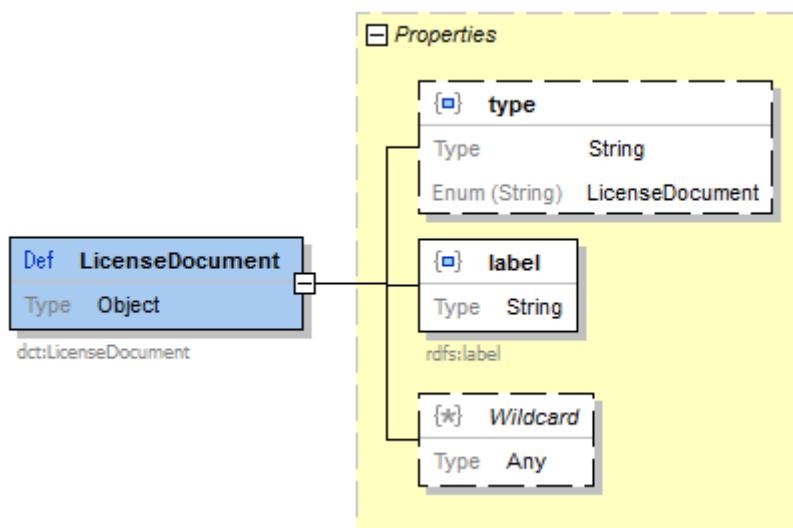


Figure 25. LicenseDocument Schema

The complete description of LicenseDocument is given in [Table 19](#).

Table 19. LicenseDocument object properties

JSON Property	Definition	Data type and values	Multiplicity and use
type \$.properties.license[*].type	Type of the element. This property has the fixed value "LicenseDocument".	Property Domain: LicenseDocument Range: String	Zero or one (optional)
label \$.properties.license[*].label	Free text content of the corresponding ISO19139-2 metadata property gmd:useLimitation.	Property Domain: LicenseDocument Range: String	One (mandatory)

RightsStatement

The proposed encoding is aligned with [\[GeoDCAT-AP\]](#) §II.15 and [\[DCMI\]](#).

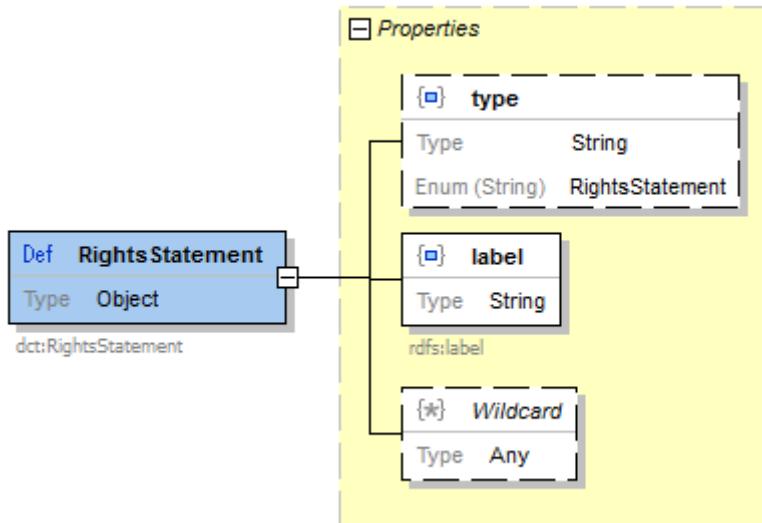


Figure 26. RightsStatement Schema

The complete description of RightsStatement is given in [Table 20](#).

Table 20. RightsStatement object properties

JSON Property	Definition	Data type and values	Multiplicity and use
type \$.properties.accessRights[*].type	Type of the element. This property has the fixed value "RightsStatement".	Property Domain: RightsStatement Range: String	Zero or one (optional)
label \$.properties.accessRights[*].label	Free text content of the corresponding ISO19139-2 metadata properties gmd:accessConstraints and gmd:otherConstraints.	Property Domain: RightsStatement Range: String	One (mandatory)

Use Constraints

Use Constraints shall be implemented through the LicenseDocument ([Table 19](#)) object.

Use constraints encoding example

```
{
  "license": [
    {
      "type": "LicenseDocument",
      "label": "Fast Registration with immediate access Data is available via EOLI-SA upon registration. For information on access and use of the On-The-Fly service please refer to the OTF FAQ page and news of 28 July 2016."
    }
  ]
}
```

Access Constraints

Access Constraints shall be implemented through the RightsStatement ([Table 20](#)) object.

Access constraints encoding example

```
{  
  "accessRights": [  
    {  
      "type": "RightsStatement",  
      "label": "Utilisation of this application is subject to ESA's Earth  
Observation Terms and Conditions"  
    }  
  ]  
}
```

7.1.6. Requirements class: Descriptive Keywords

The DescriptiveKeywords properties are inherited by the Properties block.

[GeoDCAT-AP] §II.8.2 models keyword information from services differently than keywords for datasets as [DCAT] did not cover services. This is solved in [DCAT-2018], allowing the same representations (dcat:theme and dcat:keyword) to be used for keywords in service and collection (dataset series in [GeoDCAT-AP] metadata. To align with [GeoDCAT-AP] §II.8.1, three kinds of keywords are distinguished.

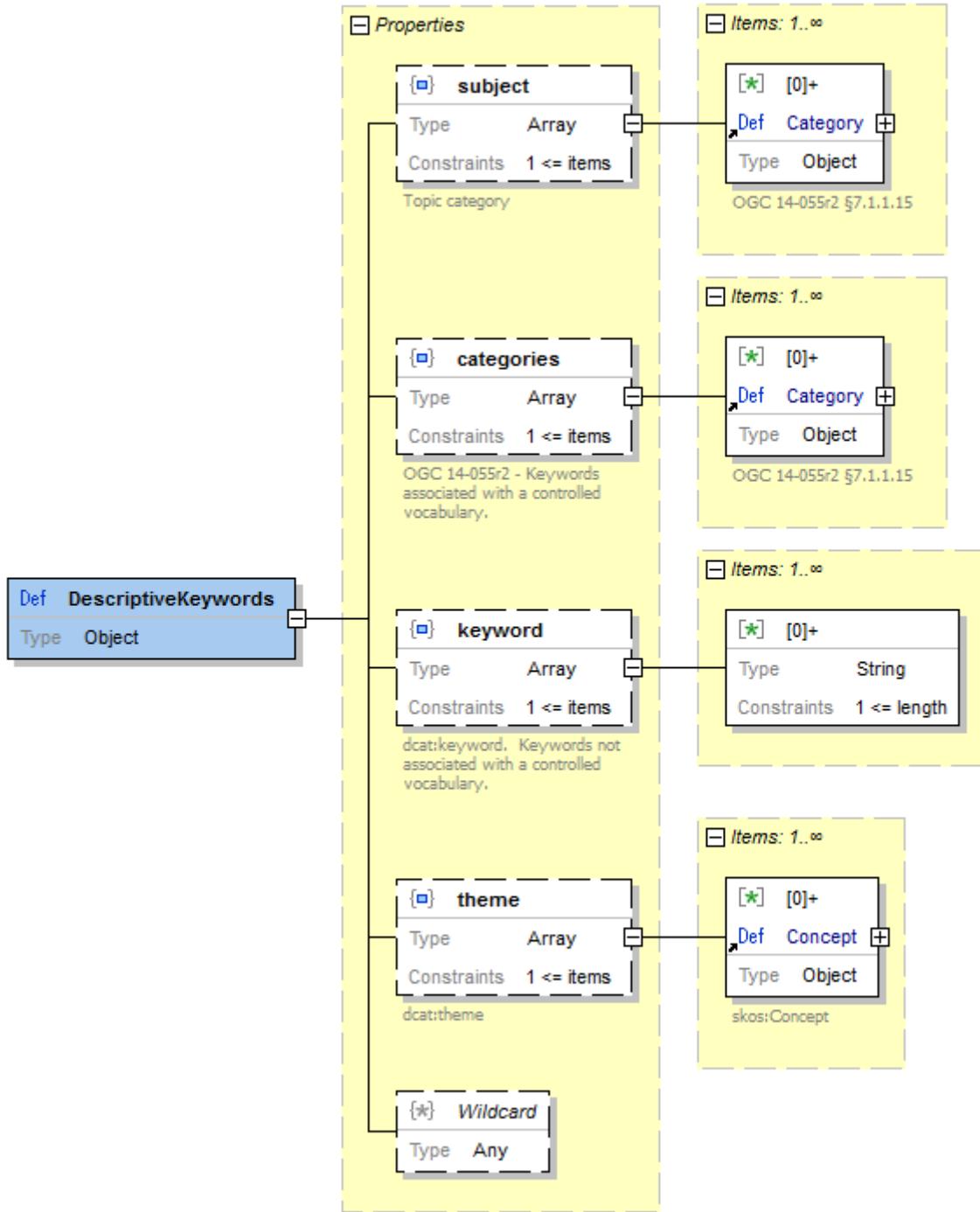


Figure 27. DescriptiveKeywords Schema

The complete description of DescriptiveKeywords is given in [Table 21](#).

Table 21. DescriptiveKeywords object properties

JSON Property	Definition	Data type and values	Multiplicity and use
subject \$.properties.subject	ISO topic categories related to the service or application. Defined in [GeoDCAT-AP] .	Range: Array of Category (Table 22)	Zero or one (optional)
categories \$.properties.categories	Keywords belonging to a controlled vocabulary related to the service or application. Defined in [OGC14-055r2] .	Range: Array of Category (Table 22)	Zero or one (optional)

JSON Property	Definition	Data type and values	Multiplicity and use
keyword \$.properties.keyword	Free keywords not belonging to a controlled vocabulary related to the service or application. Defined for dcat:DataService in [DCAT-2018] .	Range: Array of String	Zero or one (optional)
theme \$.properties.theme	Keywords belonging to a controlled vocabulary related to the service or application. Defined for dcat:DataService in [DCAT-2018] .	Range: Array of Concept (Table 22).	Zero or one (optional)

DescriptiveKeywords encoding example

```
{  
  "categories": [  
    {  
      "term": "http://www.eionet.europa.eu/gemet/concept/3650",  
      "scheme": "http://www.eionet.europa.eu/gemet/",  
      "label": "Geology"  
    },  
    {  
      "term": "https://gcmdservices.gsfc.nasa.gov/kms/concept/03f0c0a3-04a7-4ef8-8ec0-3c2266510815",  
      "scheme":  
        "https://gcmdservices.gsfc.nasa.gov/kms/concepts/concept_scheme/sciencekeyword",  
        "label": "VISIBLE IMAGERY"  
    },  
    {  
      "term": "https://gcmdservices.gsfc.nasa.gov/kms/concept/98dc8278-fe0a-4e36-a638-9d7a5b0ed826",  
      "scheme":  
        "https://gcmdservices.gsfc.nasa.gov/kms/concepts/concept_scheme/projects",  
        "label": "FedEO"  
    },  
  ],  
  "keyword": [  
    "NDVI"  
  ],  
  "subject": [  
    {  
      "term": "http://inspire.ec.europa.eu/metadata-codelist/TopicCategory/geoscientificInformation",  
      "label": "Geoscientific Information"  
    },  
    {  
      "term": "https://gcmdservices.gsfc.nasa.gov/kms/concept/d9cd5b7e-e9e7-4746-bbc8-bc69f7b606c7",  
      "label": "GEOSCIENTIFIC INFORMATION",  
      "scheme":  
        "https://gcmdservices.gsfc.nasa.gov/kms/concepts/concept_scheme/isotopiccategory"  
    }  
  ],  
  "theme": [  
    {  
      "id": "https://gcmdservices.gsfc.nasa.gov/kms/concept/03f0c0a3-04a7-4ef8-8ec0-3c2266510815",  
      "inScheme":  
        "https://gcmdservices.gsfc.nasa.gov/kms/concepts/concept_scheme/sciencekeyword",  
        "prefLabel": "VISIBLE IMAGERY"  
    }  
  ]  
}
```

Category

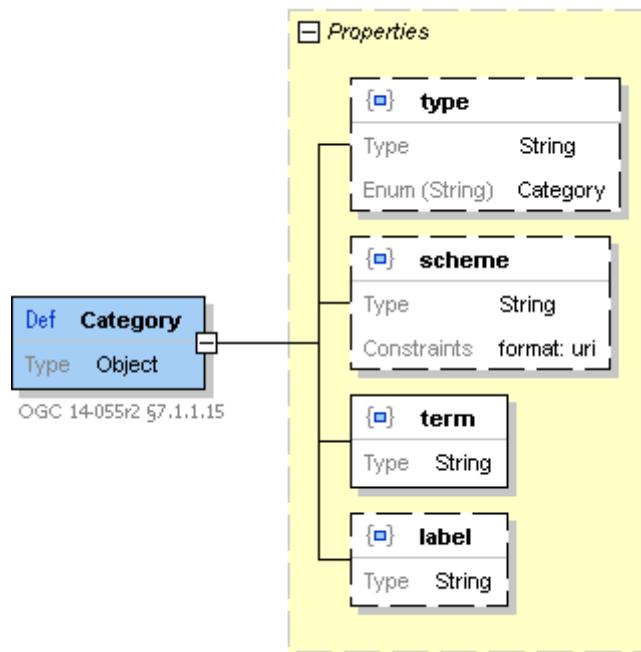


Figure 28. Category Schema

The complete description of Category is given in [Table 22](#).

Table 22. Category object properties

JSON Property	Definition	Data type and values	Multiplicity and use
type \$.properties.categories[*].type	Type of the element. This property has the fixed value "Category".	Domain: Category Range: String	Zero or one (optional)
scheme \$.properties.categories[*].scheme	Identification of the code-list defining the keyword. Defined in section 7.1.1.15 of [OGC14-055r2] .	Domain: Category Range: String (URI)	Zero or one (optional)
term \$.properties.categories[*].term	Identification of the keyword. See also OGC 08-167r2 for using of URI as string value.	Domain: Category Range: String	One (mandatory)
label \$.properties.categories[*].label	Human readable representation of the keyword.	Domain: Category Range: String	Zero or one (optional)

Category encoding example

```
{
    "type": "Category",
    "term": "http://www.eionet.europa.eu/gemet/concept/3650",
    "scheme": "http://www.eionet.europa.eu/gemet/",
    "label": "Geology"
}
```

Concept

7.1.7. Requirements class: Related URL

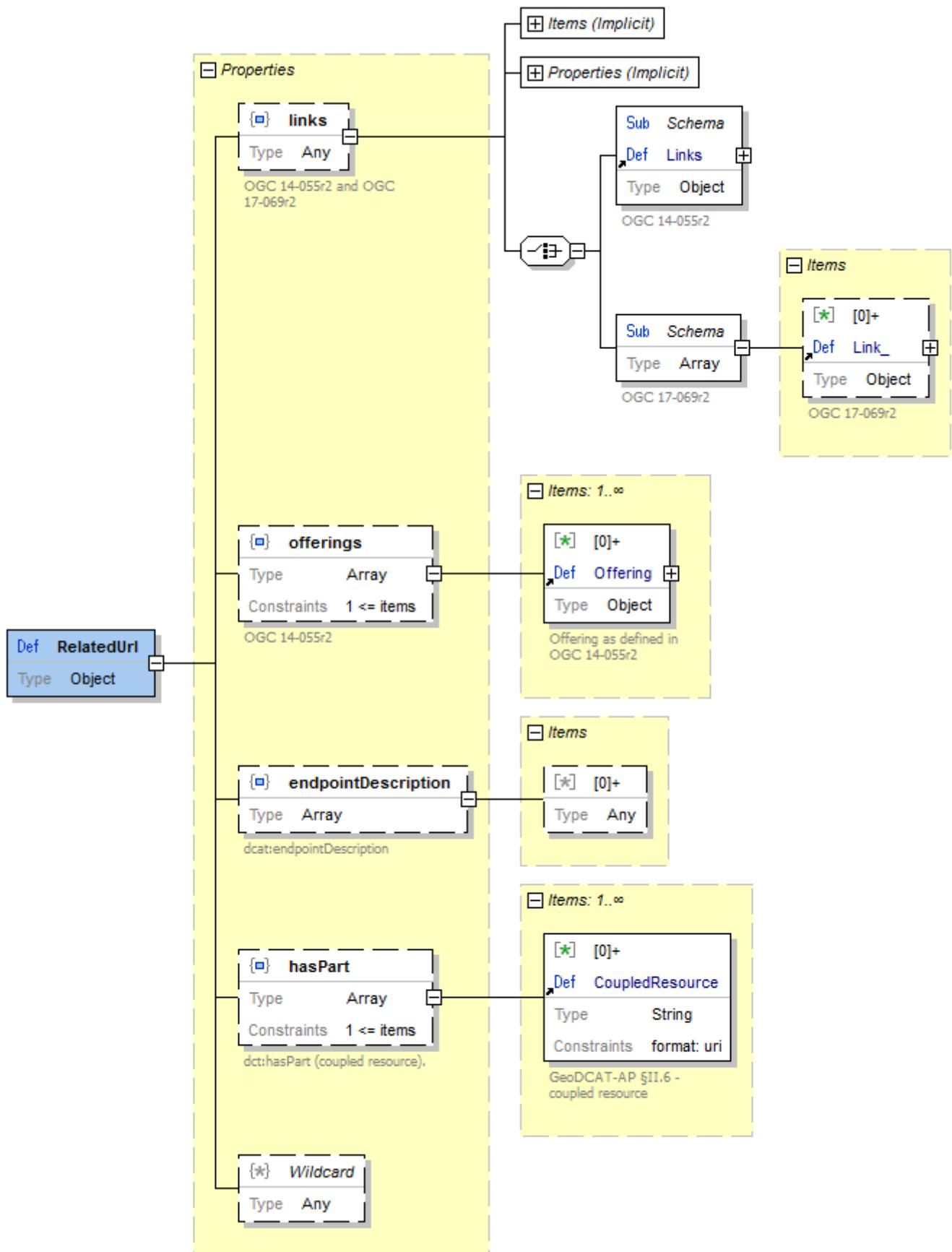


Figure 29. `RelatedUrl` Schema

The complete description of `RelatedUrl` is given in [Table 23](#). Note that the encoding of the `links`

property differs depending on the applicable service binding specification.

Table 23. RelatedUrl object properties

JSON Property	Definition	Data type and values	Multiplicity and use
links \$.properties.links	Refers to related, actionable resources including alternative metadata. Property defined in [OGC14-055r2] .	Range: Links (Table 24)	Zero or one (optional)
links \$.properties.links	Refers to related, actionable resources including alternative metadata. Property defined in [OGC17-069r2] (OGC API Features).	Range: Array of Link_ (Table 26)	Zero or one (optional)
offerings \$.properties.offerings	Service or online content offering for the resource targeted at OGC compliant clients. See [OGC14-055r2] .	Range: Array of Offering (Table 27)	Zero or one (optional)
hasPart \$.properties.hasPart	Property to link a service to the target datasets or collections. Defined in [GeoDCAT-AP] §II.6 (dct:hasPart).	Range: Array of String (URI)	Zero or more (optional)

RelatedUrl encoding example (OGC14-055r2)

```
{
  "hasPart": [
    "http://fedeo.esa.int/series/EOP:ESA:Sentinel-2"
  ],
  "links": {
    "profiles": [ ... ],
    "alternates": [ ... ],
    "describedby": [ ... ]
  },
  "offerings": [ ... ]
}
```

When the Data Model is used in combination with the OGC API-Features, the `links` property has to be encoded as per [\[OGC17-069r2\]](#) as a foreign member `<xy>.links` and is not a subproperty `<xy>.properties.links` as in [\[OGC14-055r2\]](#).

RelatedUrl encoding example (OGC17-069r2)

```
{  
  "type": "Feature",  
  
  "properties": {  
    "hasPart": [  
      "http://fedeo.esa.int/series/EOP:ESA:Sentinel-2"  
    ],  
    "offerings": [ ... ]  
  
  },  
  
  "links": [  
    { "rel": "profile", ... },  
    { "rel": "alternate", ... },  
    { "rel": "describedby", ... },  
    ...  
  ]  
}
```

Links

The Links block contains references to related resources as hypermedia links. They include specification references and references to alternative representations of the metadata. The GeoJSON encoding of the Links object uses the property names adopted by section 7.1.2 of the GeoJSON encoding for OWS Context [\[OGC14-055r2\]](#) or "link relation types" as defined in [RFC8288](#) [<https://tools.ietf.org/html/rfc8288>].

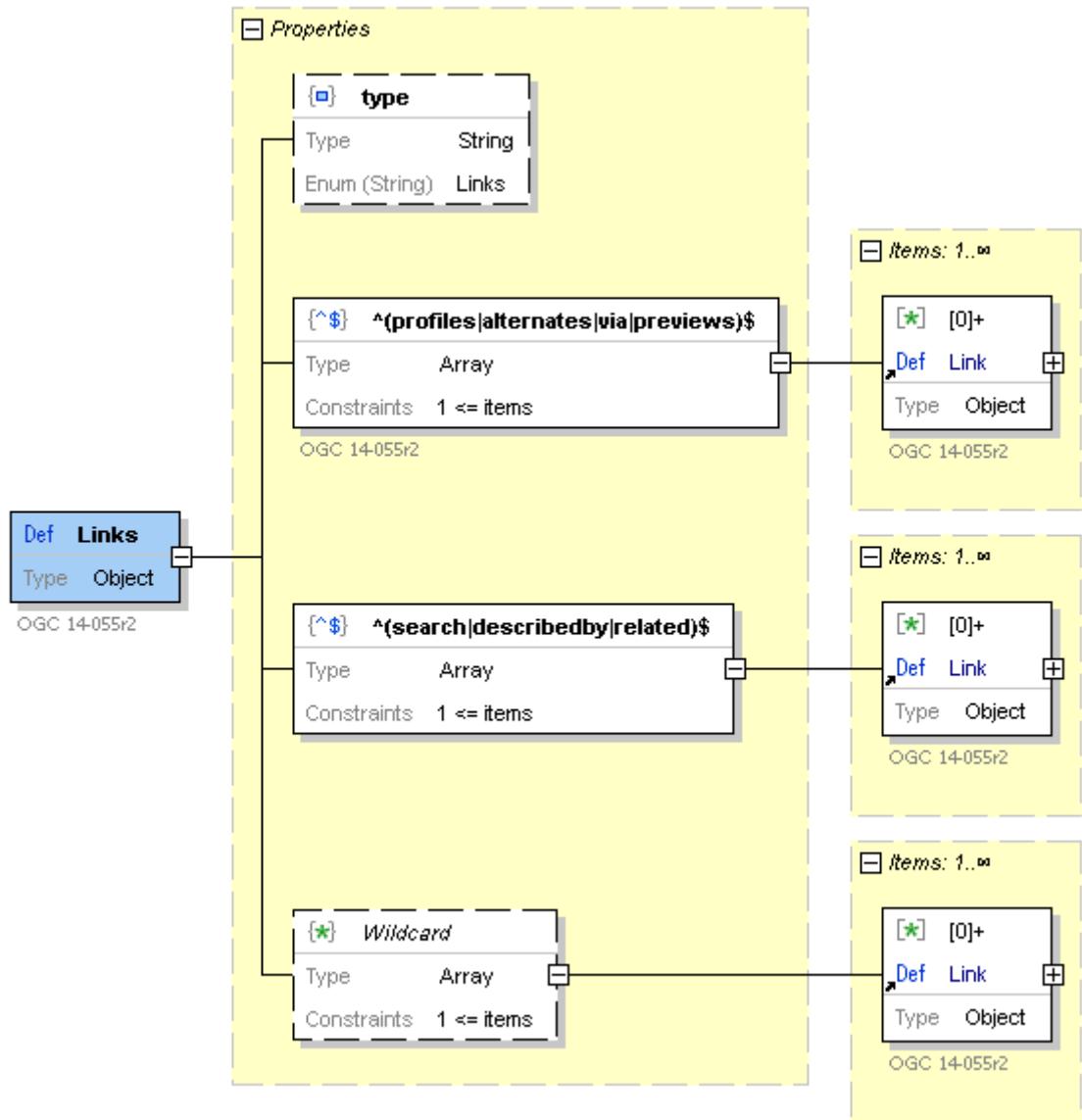


Figure 30. Links Schema

The Links object is a map containing descriptions of potential Links. The key is a "link relation type" as defined in [RFC8288](https://tools.ietf.org/html/rfc8288) [<https://tools.ietf.org/html/rfc8288>] or a relation defined in [\[OGC14-055r2\]](#). The Link objects are grouped in arrays according to the "relation" they represent. Implementers may add additional properties if additional "relations" need to be represented. For consistency, all relations refer to arrays of Link objects and not to single Link objects. This allows to have similar references with different media types included for the same relation. The general pattern is thus `$.properties.links.relation[*].href`. It should be noted that also in Atom, multiple Atom links are allowed by the response syntax, whatever the value of the "rel" attribute. Clients shall use the combination of "rel" and "type" attribute to select the proper Link.

A transactional WPS server (WPS-T) and an application package may be represented as two different resources in the catalog. The [hosts link relation](#) [<https://www.iana.org/assignments/link-relations/link-relations.xml>] can be used to refer to the catalog record representing the application package from the catalog record representing the WPS-T server.

The complete description of Links is given in [Table 24](#).

Table 24. Links object properties

JSON Property	Definition	Data type and values	Multiplicity and use
type \$.properties.links.type	Type of the element. This property has the fixed value "Links".	Range: String Fixed value: Links	Zero or one (optional)
alternates \$.properties.links.alternates	Reference to a description of the resource in an alternative format (rel="alternate"). Property defined in [OGC14-055r2] §7.1.2.	Range: array of Link (Table 25)	Zero or one (optional)
describedby \$.properties.links.describedby	Reference to a description of the resource or documentation or (human readable) information related to the resource. Such link can also be used to associate the response document with the corresponding JSON Schema file.	Range: array of Link (Table 25)	Zero or one (optional)
enclosure \$.properties.links.enclosure	Reference to the download location for the resource.	Range: array of Link (Table 25)	Zero or one (optional)
hosts \$.properties.links.hosts	Refers to the record representing the application package from the catalog record describing the WPS-T service hosting the corresponding service, i.e. on which the application package has been deployed.	Range: array of Link (Table 25)	Zero or one (optional)
previews \$.properties.links.previews	Reference to a quick-look or browse image representing the resource (rel="icon"). Property defined in [OGC14-055r2] §7.1.2.10.	Range: array of Link (Table 25)	Zero or one (optional)
profiles \$.properties.links.profiles	Specification references identifying which specifications the metadata document complies with. Property defined in [OGC14-055r2] §7.1.1.1.	Range: array of Link (Table 25)	Zero or one (optional)
related \$.properties.links.related	Reference to a related resource. Is also used to return results of correlated search as per [OGC13-068] .	Range: array of Link (Table 25)	Zero or one (optional)
via \$.properties.links.via	Reference to a description of the resource of which the response document is derived (e.g. source of information). Property defined in [OGC14-055r2] §7.1.1.14.	Range: array of Link (Table 25)	Zero or one (optional)

hosts *Link encoding example (OGC API Processes)*

```
{  
  "id": "http://fedeo.esa.int/services/org.52north.dev.testbed.javaps-eopad.rest",  
  "type": "Feature",  
  "geometry": null,  
  "properties": {  
    "identifier": "org.52north.dev.testbed.javaps-eopad.rest",  
    "kind": "http://inspire.ec.europa.eu/metadata-codelist/ResourceType/service",  
    "title": "Processing API at testbed.dev.52north.org",  
    ...  
  },  
  "links": {  
    "profiles": [  
      {  
        "href": "http://www.opengis.net/spec/owc-geojson/1.0/req/core"  
      },  
      {  
        "href": "http://www.opengis.net/spec/eopad-geojson/1.0/req/core"  
      }  
    ],  
    "hosts": [  
      {  
        "href":  
          "http://fedeo.esa.int/services/org.n52.project.tb15.eopad.NdviCalculation",  
          "type": "application/json",  
          "title": "Application Package description"  
        },  
        {  
          "href":  
            "http://fedeo.esa.int/services/org.n52.project.tb15.eopad.SentinelQualityAreaOfInterest",  
            "type": "application/json",  
            "title": "Application Package description"  
          }  
    ]  
  }  
}
```

Link

The Link block contains the properties of a hypermedia link to a resource identified by its URI [\[OGC14-055r2\]](#).

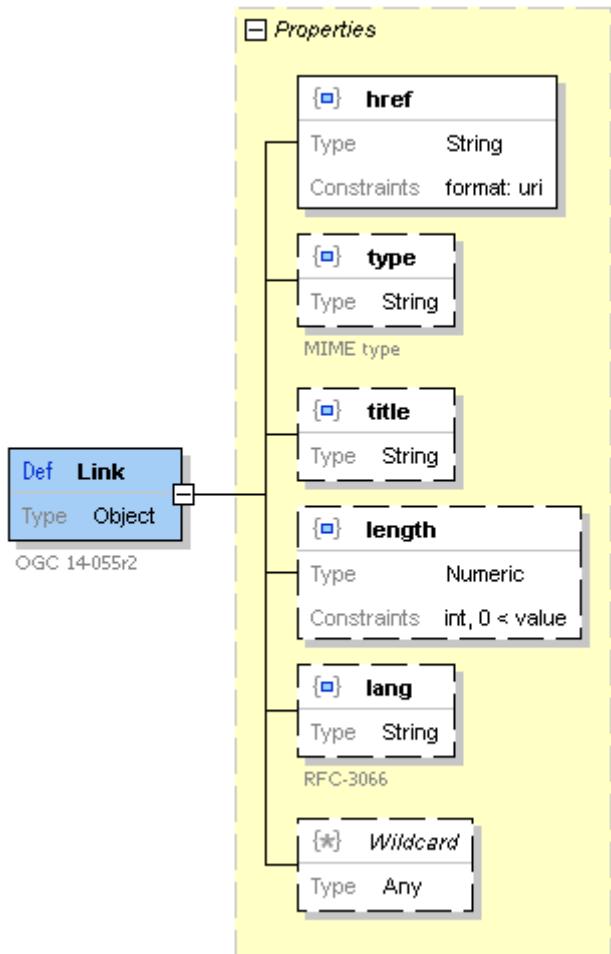


Figure 31. Link Schema

The complete description of Link is given in Table 25.

The GeoJSON encoding of each Link object uses the encoding defined in section 7.1.10 of the GeoJSON encoding for OWS Context [OGC14-055r2]. The equivalent JSON-LD encoding is consistent with section 6.1.2.2 of OGC 15-053.

Table 25. Link object properties

JSON Property	Definition	Data type and values	Multiplicity and use
href \$.properties.links.[].href	URI describing the related resource. Property defined in [OGC14-055r2] §7.1.10.	Domain: Link Range: String (URI)	One (mandatory)
type \$.properties.links.[].type	Hint about the type of the representation that is expected to be returned when the value of href is dereferenced. Property defined in [OGC14-055r2] §7.1.10.	Domain: Link Range: String (contains a media type)	Zero or one (optional)
title \$.properties.links.[].title	Human readable information about the link. Property defined in [OGC14-055r2] §7.1.10.	Domain: Link Range: String	Zero or one (optional)

JSON Property	Definition	Data type and values	Multiplicity and use
length \$.properties.links[0].length	Hint about the content length (in bytes) of the representation that is expected to be returned when the value of href is dereferenced. Property defined in [OGC14-055r2] §7.1.10.	Domain: Link Range: Integer	Zero or one (optional)
lang \$.properties.links[0].lang	Language of the resource pointed to by the href attribute. String type, not empty with an RFC-3066 code. Property defined in [OGC14-055r2] §7.1.10.	Domain: Link Range: String	Zero or one (optional)

An alternative representation is required when the Data Model is used in combination with an OGC API Features service binding.

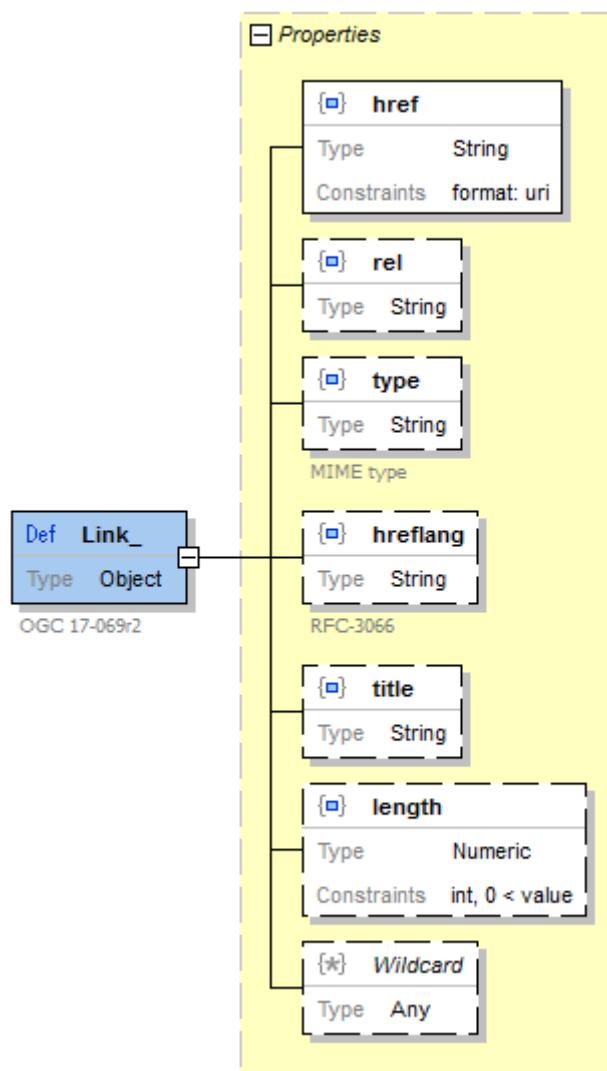


Figure 32. Link_ Schema

Table 26. Link_ object properties

JSON Property	Definition	Data type and values	Multiplicity and use
href \$.properties.links[0].href	URI describing the related resource. Property defined in [OGC17-069r2] .	Domain: Link Range: String (URI)	One (mandatory)

JSON Property	Definition	Data type and values	Multiplicity and use
rel \$.properties.links[*].rel	Link relation type as defined in RFC8288 [https://tools.ietf.org/html/rfc8288]. Property defined in [OGC17-069r2] .	Domain: Link Range: String (URI)	Zero or one (optional)
type \$.properties.links[*].type	Hint about the type of the representation that is expected to be returned when the value of href is dereferenced. Property defined in <OGC17-069r2>.	Domain: Link Range: String (contains a media type)	Zero or one (optional)
title \$.properties.links[*].title	Human readable information about the link. Property defined in <OGC17-069r2>.	Domain: Link Range: String	Zero or one (optional)
length \$.properties.links[*].length	Hint about the content length (in bytes) of the representation that is expected to be returned when the value of href is dereferenced. Property defined in <OGC17-069r2>.	Domain: Link Range: Integer	Zero or one (optional)
hreflang \$.properties.links[*].hreflang	Language of the resource pointed to by the href attribute. String type, not empty with an RFC-3066 code. Property defined in <OGC17-069r2>.	Domain: Link Range: String	Zero or one (optional)

A possible JSON-LD encoding for the above alternative Link encoding could be:

Alternative Link encoding in JSON-LD

```
"dcat:qualifiedRelation": [
  {
    "@type": "dcat:Relationship",
    "dct:format": "application/vnd.iso.19139-2+xml",
    "dct:language": {
      "@id": "http://id.loc.gov/vocabulary/iso639-1/en"
    },
    "dct:relation": "http://fedeo.esa.int/services/NdviCalculation?httpAccept=application/vnd.iso.19139-2%2Bxml",
    "dct:title": "ISO 19139-2 metadata",
    "dcat:hadRole": {
      "@id": "http://www.iana.org/assignments/relation/alternate"
    }
  },
  ...
]
```

This requires the following changes to the normative JSON-LD @context.

Alternative Link encoding in JSON-LD @context

```
"links": {  
    "@id": "dcat:qualifiedRelation",  
    "@context": {  
        "@base": "http://www.iana.org/assignments/relation/",  
        "Link": "dcat:Relationship",  
        "rel": {  
            "@id": "dcat:hadRole",  
            "@type": "@id"  
        },  
        "type": "dct:format",  
        "href": "dct:relation"  
    }  
}
```

Offering

The Offering block is defined in [OGC 14-055r2](#). This block describes a service endpoint or inline content offering for the EO service/application and is intended to be consumed by clients that support OGC standards. In Testbed-15, the Offering block is also used to describe some non-OGC endpoints and offerings.

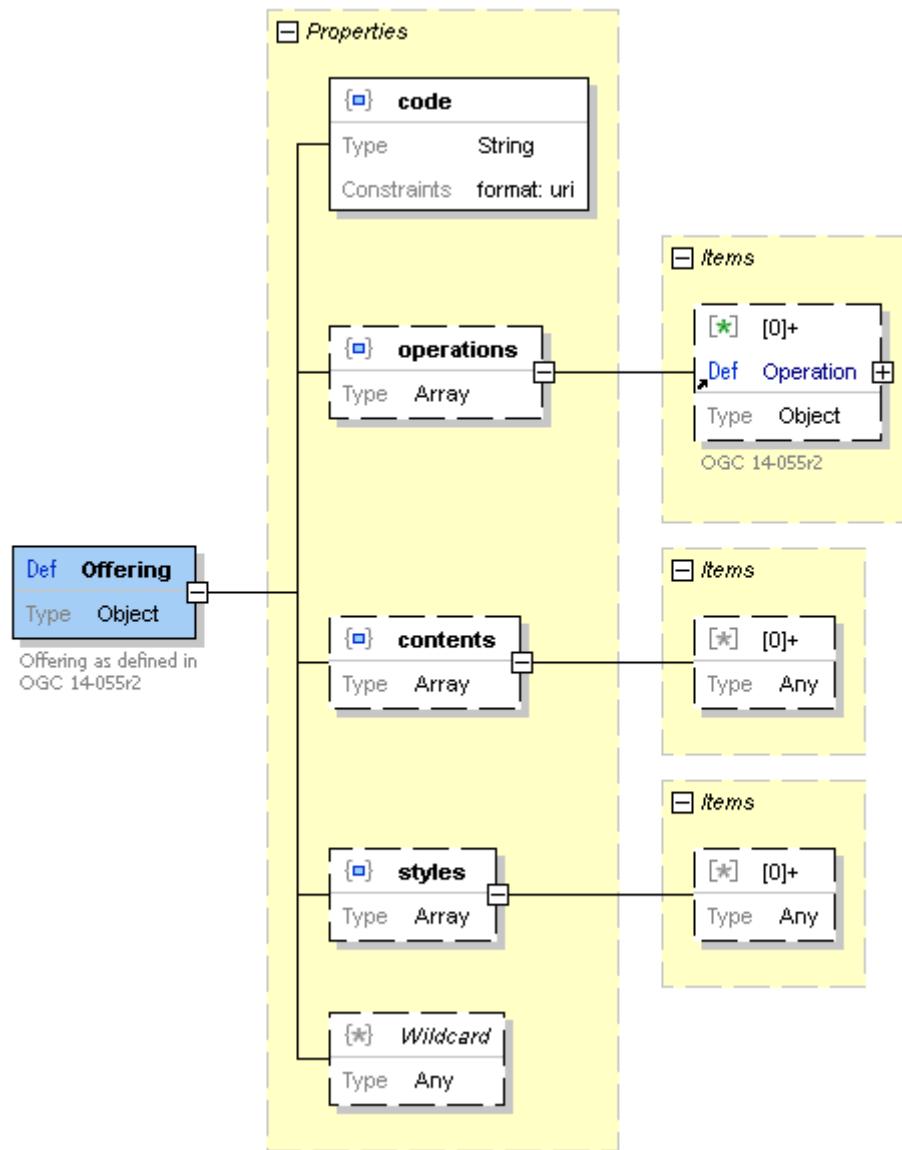


Figure 33. Offering Schema

In the context of the current ER, an Offering can be used to describe:

- View services allowing visualization via OGC Web Map Service (WMS) or Web Map Tiling Service (WMTS) interfaces,
- Reference to download or ordering endpoints according to specific OGC protocols e.g. Web Coverage Service (WCS),
- Reference to processing endpoints including Web Processing Service (WPS) or hosted processing services (WPS-T).

These are covered "out-of-the-box" using OGC 14-055r2. Testbed-15 has also identified additional needs which include:

- Application available as a deployable Docker container.
- Application available as application package (i.e. Process Description)
- Reference to an OpenDAP endpoint.

The complete description of Offering is given in [Table 27](#).

Table 27. Offering object properties

JSON Property	Definition	Data type and values	Multiplicity and use
type \$.properties.offerings[*].type	Type of the element. This property has the fixed value "Offering".	Domain: Offering Range: String	Zero or one (optional)
code \$.properties.offerings[*].code	URI identifying the type of offering, e.g. requirement class identifier (URI) defining the offering listed in §7.2 to 7.12 of [OGC14-055r2]. Is defined by [OGC14-055r2] §7.1.3.1.	Domain: Offering Range: String (URI)	One (mandatory)
operations \$.properties.offerings[*].operations	Array of operations used to invoke the service. Is defined by [OGC14-055r2] §7.1.3.2.	Domain: Offering Range: array of Operation	Zero or more (optional)
contents \$.properties.offerings[*].contents	Array of contents (inline or by reference). Is defined by [OGC14-055r2] §7.1.3.3 and §7.1.5.	Domain: Offering Range: array of Object	Zero or more (optional)
styles \$.properties.offerings[*].styles	Array of style sets. Is defined by [OGC14-055r2] §7.1.3.4 and §7.1.6.	Domain: Offering Range: array of Object	Zero or more (optional)

Offering encoding example for a WMS endpoint

```
{
  "type": "Offering",
  "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wms",
  "operations": [
    {
      "code": "GetCapabilities",
      "method": "GET",
      "type": "application/xml",
      "href": "http://eumetview.eumetsat.int/geoserv/wms?REQUEST=GetCapabilities&version=1.3.0&service=WMS"
    }
  ]
}
```

Offering encoding example for a WPS instance

```
{  
  "type": "Offering",  
  "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wps",  
  "operations": [  
    {  
      "code": "GetCapabilities",  
      "method": "GET",  
      "type": "application/xml",  
      "href": "  
http://tep.esa.int/wps/processing?REQUEST=GetCapabilities&SERVICE=WPS",  
      "result": {  
        "type": "application/xml",  
        "content": "<wps:Capabilities>...</wps:Capabilities>"  
      }  
    }  
  ]  
}
```

The offering code is defined in [OGC 14-055r2](#) as the requirement class identifier (URI) for the extension defining the offering/operation. “code” can be an OGC Web Services Context (OWC) extension or one defined in a profile. Testbed-15 defined a number of new offering codes to cover its different use cases. These are identified as

```
http://www.opengis.net/spec/eopad-geojson/1.0/req/
```

Offering encoding example (Docker image)

```
{  
  "type": "Offering",  
  "code": "http://www.opengis.net/spec/eopad-geojson/1.0/req/docker/image",  
  "contents": [  
    {  
      "type": "text/plain",  
      "content": "docker.52north.org/eopad/snap-sentinel-2-ndvi:latest"  
    }  
  ]  
}
```

In the above example, the *content* property corresponds to a reference to the Docker image. This property shall not include any protocol and includes an optional registry host name, an optional username, the image ID and then an optional tag (i.e. syntax used by the [Docker engine REST API](#) [<https://docs.docker.com/engine/api/v1.37/>]):

```
[registry.host.name/]NAME[:TAG|@DIGEST]
```

[Seed](https://ngageoint.github.io/seed/seed.html) [https://ngageoint.github.io/seed/seed.html] compliant Docker images can be described as offerings as shown below. In this case, the reference to the Docker image is derived from the information in the manifest: <name>-<jobVersion>-seed:<packageVersion>.

Offering encoding example (Seed-compliant Docker image)

```
{  
  "type": "Offering",  
  "code": "http://www.opengis.net/spec/eopad-geojson/1.0/req/seed/manifest",  
  "contents": [  
    {  
      "type": "application/json",  
      "content": {  
        "seedVersion": "1.0.0",  
        "job": {  
          "name": "MultiSensorNDVI",  
          "jobVersion": "1.0.0",  
          "packageVersion": "1.0.0",  
          "title": "Multi Sensor NDVI",  
          "description": "NDVI is calculated after the two bands values Near Infrared and red. It is calculated by this formula : NDVI = (NIR-Red)/(NIR+Red)",  
          "timeout": 3600,  
          "maintainer": {  
            "name": "ESA/ESRIN",  
            "email": "eohelp@eo.esa.int"  
          },  
          "resources": {},  
          "interface": {},  
          "errors": []  
        }  
      }  
    }  
  ]  
}
```

In the following example, the *content* property of the **DescribeProcess** operation *result* provides access to the Process Description object for the application which is the recommended encoding for the Application Package. It is shortened for readability (See [Annex C.2](#) for the complete application package). The Process Description defines the *inputs* and *outputs* of the application. It is the main part of the Application Package encoding defined in [\[OGC18-049r1\]](#) and [\[OGC18-050r2\]](#).

Offering encoding example for a WPS DescribeProcess call

```
{  
  "type": "Offering",  
  "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wps",  
  "operations": [  
    {  
      "code": "DescribeProcess",  
      "method": "GET",  
      "type": "application/json",  
      "href": "http://52north.org/eopad/wps/processes/NdviCalculation",  
      "result": {  
        "type": "application/json",  
        "content": {  
          "processDescription": {  
            "process": {  
              "id": "NdviCalculation"  
            }  
          }  
        }  
      }  
    }  
  ]  
}
```

Offering encoding example for an OGC API Processes request

```
{  
  "type": "Offering",  
  "code": "http://www.opengis.net/spec/eopad-geojson/1.0/req/ogc-api-processes",  
  "operations": [  
    {  
      "code": "LandingPage",  
      "method": "GET",  
      "href": "https://testbed.dev.52north.org/javaps-eopad/rest",  
      "type": "application/json"  
    },  
    {  
      "code": "Service",  
      "method": "GET",  
      "href": "https://testbed.dev.52north.org/javaps-eopad/rest/api/",  
      "type": "application/openapi+json;version=3.0"  
    },  
    {  
      "code": "Conformance",  
      "method": "GET",  
      "href": "https://testbed.dev.52north.org/javaps-eopad/rest/conformance/",  
      "type": "application/json"  
    },  
    {  
      "code": "Process",  
      "method": "GET",  
      "href": "https://testbed.dev.52north.org/javaps-eopad/rest/processes/  
    }  
  ]  
}
```

```

    "code": "Processes",
    "method": "GET",
    "href": "https://testbed.dev.52north.org/javaps-eopad/rest/processes/",
    "type": "application/json"
},
{
    "code": "DeployProcess",
    "method": "POST",
    "href": "https://testbed.dev.52north.org/javaps-eopad/rest/processes/",
    "type": "application/json"
},
{
    "code": "DescribeProcess",
    "method": "GET",
    "href": "https://testbed.dev.52north.org/javaps-eopad/rest/processes/org.n52.project.tb15.eopad.NdviCalculation",
    "type": "application/json",
    "result": {
        "type": "application/json",
        "content": {
            "process": {
                "id": "org.n52.project.tb15.eopad.NdviCalculation",
                "title": "Calculation of NDVI using the SNAP toolbox for
Sentinel-2",
                ...
            }
        }
    }
}
]
}

```

If a full Application Package as defined in [OGC18-049r1] needs to be embedded in the metadata, then the Offering can refer to the `DeployProcess` operation *request* instead. The (mandatory) `href` property in this case has no meaning as the Application Package may not yet be deployed and may be deployed in multiple places, and therefore the `data URI scheme` [https://en.wikipedia.org/wiki/Data_URI_scheme] is used.

Offering encoding example (Application Package)

```
{  
  "type": "Offering",  
  "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wps",  
  "operations": [  
    {  
      "code": "DeployProcess",  
      "method": "POST",  
      "type": "application/json",  
      "href": "data:",  
      "request": {  
        "type": "application/json",  
        "content": {  
          "processDescription": {  
            "process": {  
              "id": "MultiSensorNDVIStackGenerator"  
            },  
            "executionUnit": [  
              {  
                "href": "https://some-  
host/CWL/MultiSensorStackGenerator.cwl"  
              }  
            ],  
            "deploymentProfileName":  
              "http://www.opengis.net/profiles/eoc/workflow"  
          }  
        }  
      }  
    }  
  ]  
}
```

Option

NOTE The property `$.properties.endpointDescription` (i.e. `dcat:endpointDescription`) defined in [\[DCAT-2018\]](#) might be used to refer to the description of an endpoint, e.g. an OpenAPI document, OGC GetCapabilities response, OGC DescribeProcess response or OpenSearch Description Document.

Feature encoding example (with endpointDescription)

```
{  
  "type": "Feature",  
  "id": "http://fedeo.esa.int/services/NdviCalculation",  
  "bbox": [...],  
  "geometry": {...},  
  "properties": {  
    ...  
    "endpointDescription": [  
      "http://52north.org/eopad/wps/processes/NdviCalculation"  
    ]  
  }  
}
```

Feature JSON-LD encoding example (with endpointDescription)

```
{  
  "@context": {  
    "owc": "http://www.opengis.net/ont/owc/1.0/",  
    "dcat": "http://www.w3.org/ns/dcat#"  
  },  
  "@type": "dcat:DataService",  
  "dcat:endpointDescription": {  
    "@type": "owc:Offering",  
    "@id": "http://www.spacebel.be/offering/0001",  
    "owc:code": "http://www.opengis.net/spec/eopad-geojson/1.0/req/docker/image",  
    "owc:contents": [  
      {  
        "owc:type": "text/plain",  
        "owc:content": "xxx/yyy/ZZZ:latest"  
      }  
    ]  
  }  
}
```

Operation

The Operation block is fully defined in [OGC 14-055r2](#). This block describes an operation of a service or inline content offering and is intended to be consumed by OGC-compliant clients. The "code" property identifies the OGC operation name, e.g. GetCapabilities, GetMap etc.

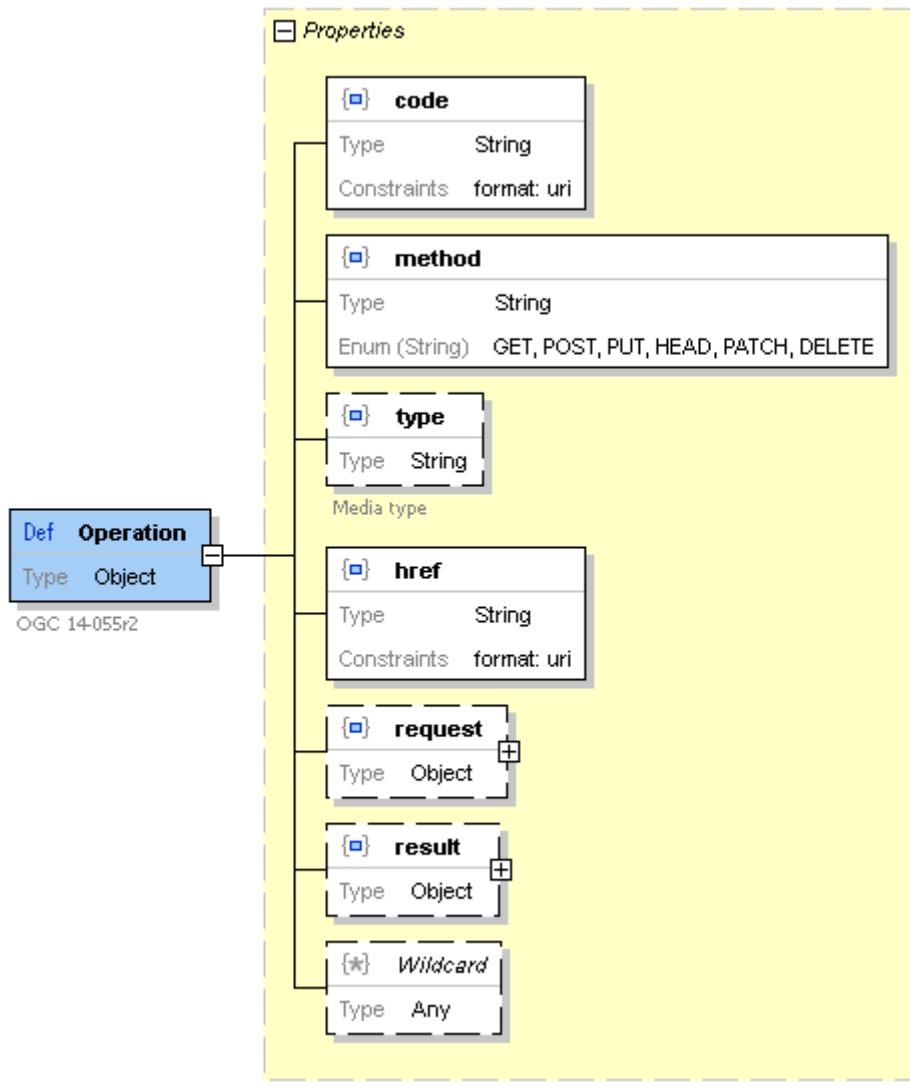


Figure 34. Operation Schema

The complete description of Operation is given in [Table 28](#).

Table 28. Operation object properties

JSON Property	Definition	Data type and values	Multiplicity and use
code \$.properties.offerings[1].operations[1].code	URI identifying the type of operation, Typically the OGC Service request type, e.g. “GetCapabilities” or “GetMap”. Is defined by [OGC14-055r2] §7.1.4.1.	Domain: Operation Range: String	One (mandatory)
method \$.properties.offerings[1].operations[1].method	Code identifying the HTTP verb type of Operation. Examples: GET, POST, PUT, HEAD, PATCH, DELETE. Is defined by [OGC14-055r2] §7.1.4.2.	Domain: Operation Range: String	One (mandatory)
type \$.properties.offerings[1].operations[1].type	MIME type of the expected results. Is defined by [OGC14-055r2] §7.1.4.3.	Domain: Operation Range: String	Zero or one (optional)

JSON Property	Definition	Data type and values	Multiplicity and use
href \$.properties.offerings[0].operations[0].href	Service request URL. Is defined by [OGC14-055r2] §7.1.4.4.	Domain: Operation Range: String (URI)	One (mandatory)
request \$.properties.offerings[0].operations[0].request	Optional request body content. Is defined by [OGC14-055r2] §7.1.4.5.	Domain: Operation Range: Object	Zero or one (optional)
result \$.properties.offerings[0].operations[0].result	Optional result payload of the operation. Is defined by [OGC14-055r2] §7.1.4.6.	Domain: Operation Range: Object	Zero or one (optional)

For examples, please refer to the Offering examples given in the previous section.

7.1.8. Requirements class: Spatial Information

Spatial Extent

Geometry

The Geometry object is defined in [\[GeoJSON\]](#) and contains a type property and coordinates property. The object can be any of the specializations described below. The expected value for the coordinates property depends on the type of the geometry.

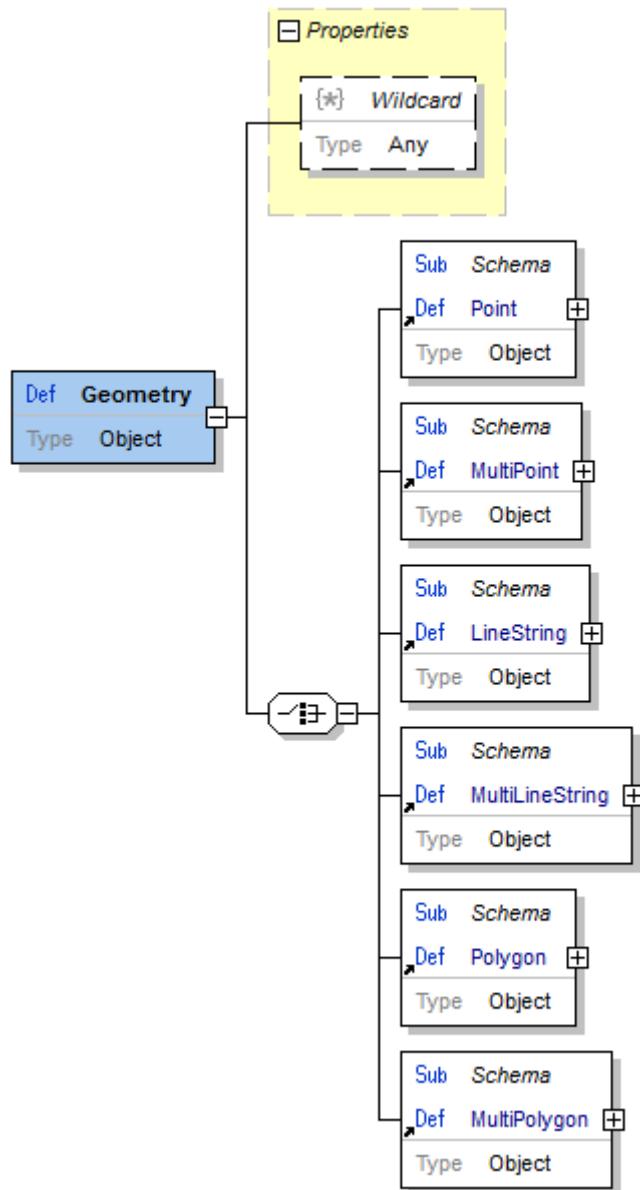


Figure 35. Geometry Schema

The complete description of the Geometry properties is given in [\[GeoJSON\]](#).

7.1.9. Requirements class: Acquisition Information

The acquisition information applicable to EO processing services and applications (See e.g. [\[UMM-S\]](#) §2.2.1.14) is a subset of the acquisition information applicable to EO products. The JSON encoding of this subset is identical to the encoding defined in [\[OGC17-003\]](#) and [\[OGC17-084\]](#). It should be noted that the forthcoming revision of the Unified Metadata Model Services (UMM-S) specification [\[UMM-S\]](#) will use collection-service association to infer this information (platform and instrument).

The AcquisitionInformation block can appear multiple times and contains information about the related platform (i.e. satellite) and the instrument.

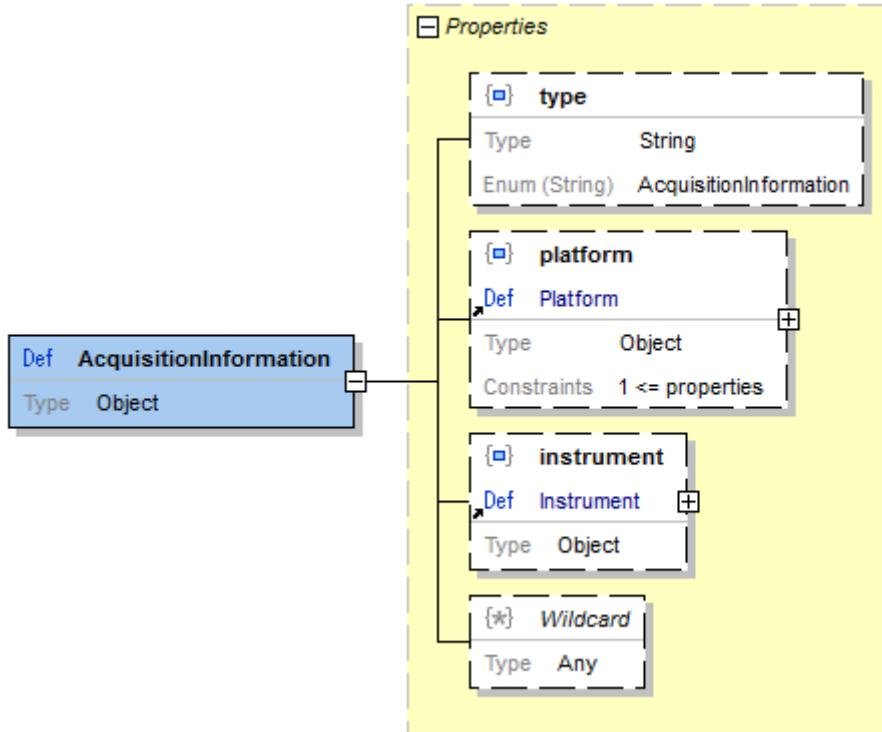


Figure 36. *AcquisitionInformation Schema*

The complete description of the *AcquisitionInformation* properties is given in [\[OGC17-003\]](#).

7.2. GeoJSON Search Response Encoding

7.2.1. Requirements class: FeatureCollection

The search interface returns search responses in [\[GeoJSON\]](#) format as defined in OGC 17-047 and depicted in [Figure 37](#). A response is represented as a FeatureCollection and each Feature represents an EO processing application/service matching the search criteria and is encoded as EOPAD metadata presented in [Section 7.2](#).

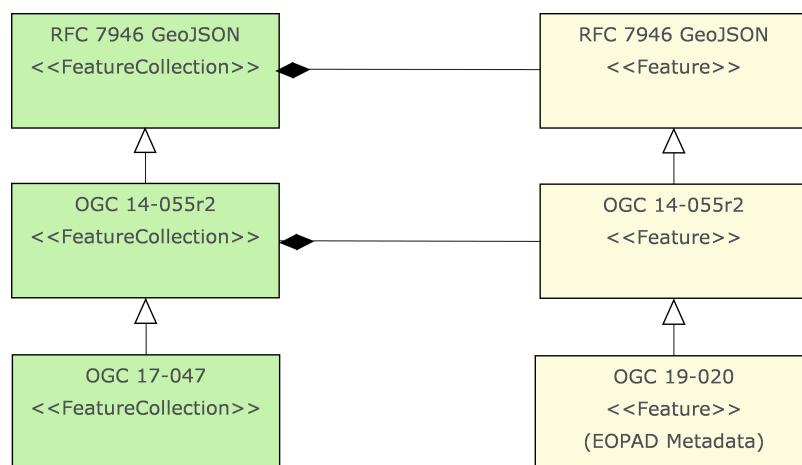


Figure 37. *GeoJSON search response structure*

Chapter 10 of [OGC 17-047](#) shows where the facetedResults fit in the GeoJSON response but left the

detailed definition of the FacetedResults object as future work. Testbed-15 defines the expected response in Annex E.2 of the current document compliant with section 8.2 of [OASIS searchRetrieve Part 3 SRU](#).

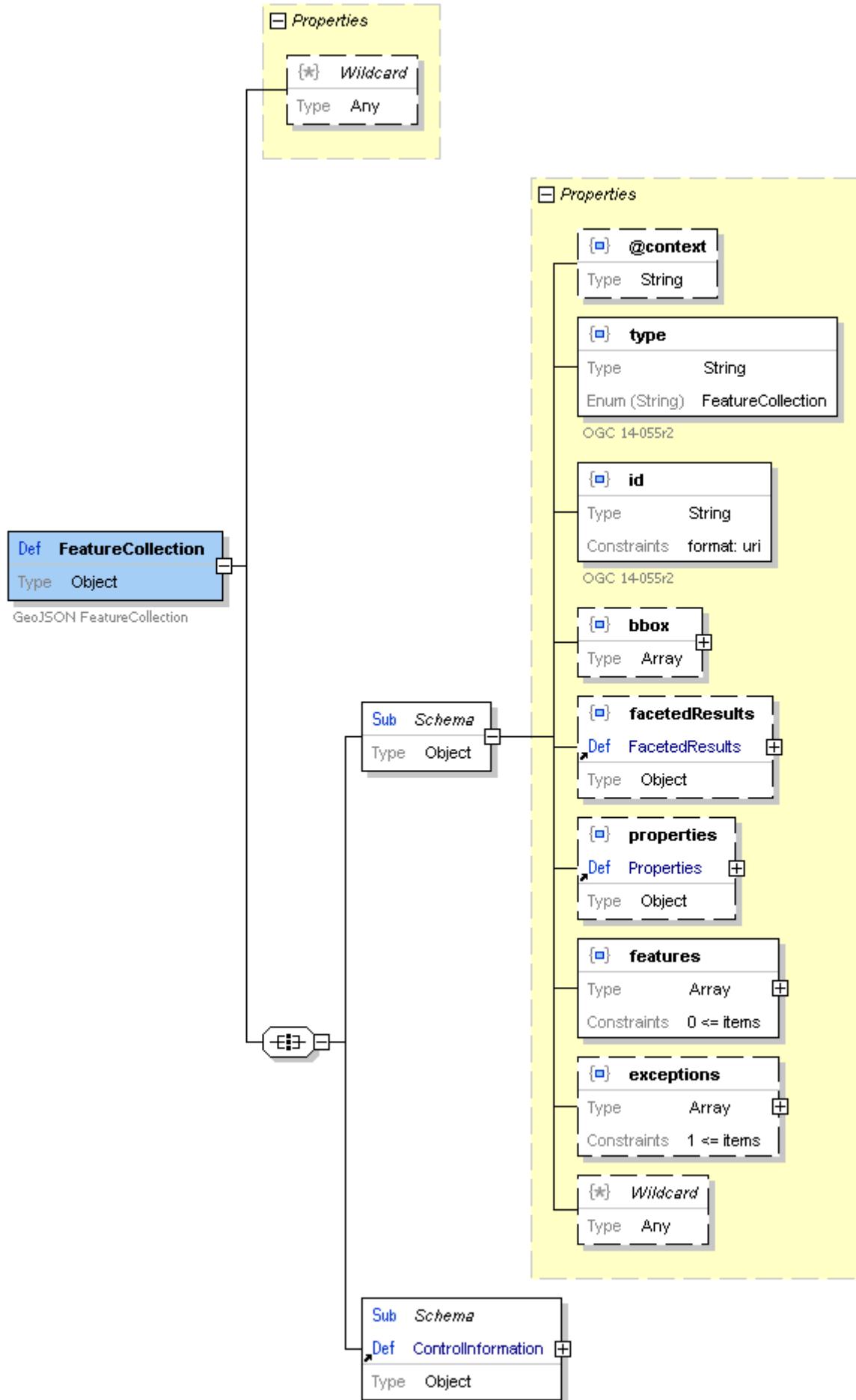


Figure 38. FeatureCollection Schema

The complete description of FeatureCollection is given in [OGC17-047]. Testbed-15 extends the

definition with the properties shown in [Table 29](#). All other properties are defined in [\[OGC17-047\]](#).

Table 29. FeatureCollection object properties

JSON Property	Definition	Data type and values	Multiplicity and use
facetedResults \$.facetedResults	Representation of the faceted results.	Property Range: FacetedResults as defined in Table 30 .	Zero or one (optional)

FeatureCollection encoding example

```
{
  "type": "FeatureCollection",
  "id": "http://fedeo.esa.int/services?startRecord=1&maximumRecords=10&query=Vegetation",
  "totalResults": 2,
  "startIndex": 1,
  "itemsPerPage": 10,
  "queries": {
    "request": [
      {
        "count": 10,
        "searchTerms": "Vegetation",
        "startIndex": 1
      }
    ]
  },
  "facetedResults": {
    "datasource": [
      ...
    ],
    "properties": {
      "title": "FEDEO Clearinghouse - Search Response",
      "creator": "FEDEO Clearinghouse",
      "updated": "2019-09-27T11:27:19Z",
      "lang": "en",
      "rights": "Copyright 2016-2019, European Space Agency",
      "links": {
        "profiles": [
          {
            "href": "http://www.opengis.net/spec/owc-geojson/1.0/req/core"
          },
          {
            "href": "http://www.opengis.net/spec/os-geojson/1.0/req/core"
          }
        ],
        ...
      }
    }
  }
}
```

```

"first": [
  {
    "href": "http://fedeo.esa.int/services?startRecord=1&maximumRecords=10&query=Vegetation",
    "type": "application/geo+json",
    "title": "first results"
  }
],
"last": [
  {
    "href": "http://fedeo.esa.int/services?startRecord=1&maximumRecords=10&query=Vegetation",
    "type": "application/geo+json",
    "title": "last results"
  }
],
"search": [
  {
    "href": "http://fedeo.esa.int/api",
    "type": "application/opensearchdescription+xml",
    "title": "search"
  }
]
},
"features": [
  {
    "type": "Feature",
    "id": "http://fedeo.esa.int/services/MultiSensorNDVI",
    ...
  },
  {
    "type": "Feature",
    "id": "http://fedeo.esa.int/services/NdviCalculation",
    ...
  }
]
}

```

FacetedResults

The server can supply faceted results for a query: i.e. an analysis of how the search results are distributed over various categories (or "facets"). For example, the analysis may reveal how the results are distributed by organization. The client might then refine the query to one particular organization among those listed.

If a catalog federates the offering of multiple (exploitation) platforms, then in the faceted results, multiple data sources may be exposed.

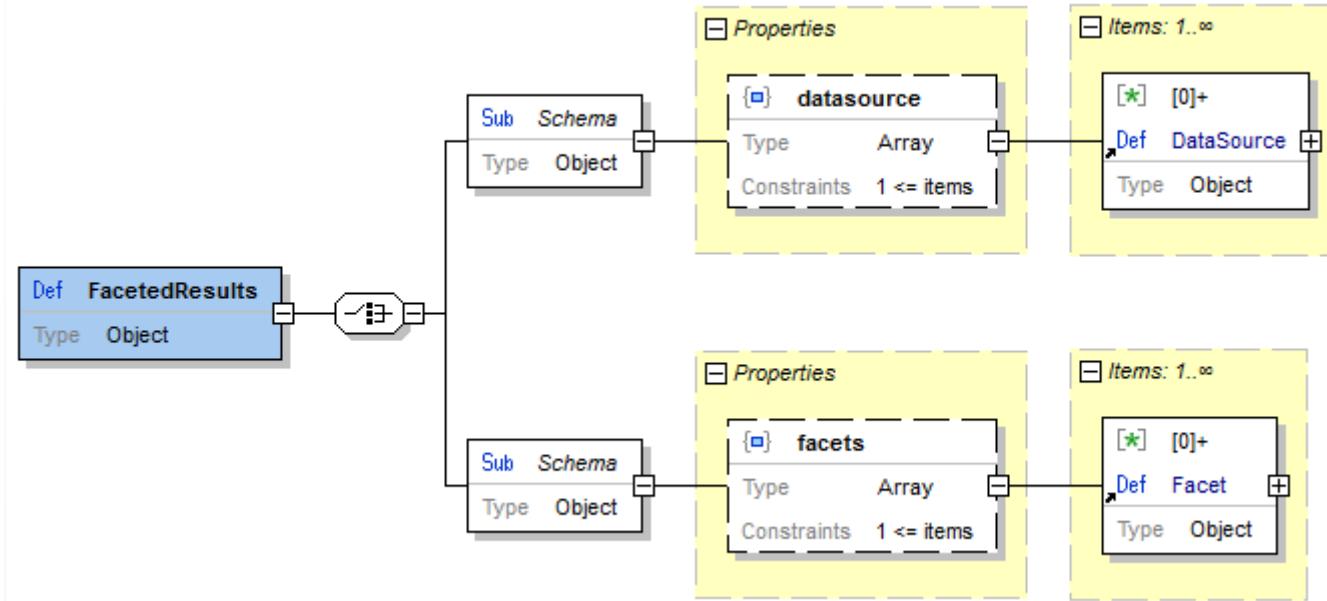


Figure 39. *FacetedResults Schema*

The complete description of *FacetedResults* is given in [Table 30](#).

Table 30. FacetedResults object properties

JSON Property	Definition	Data type and values	Multiplicity and use
type \$.facetedResults.type	Type of the object. This property has the fixed value "FacetedResults".	String Fixed value: "FacetedResults"	Zero or one (optional)
datasource \$.facetedResults.datasource	List of distributed data sources providing faceted search results and corresponding faceted search results.	Domain: FacetedResults Range: array of DataSource as defined in Table 31 .	Zero or one (optional)
facets \$.facetedResults.facets	List of facets and corresponding faceted search results.	Domain: FacetedResults Range: array of Facet as defined in Table 32 .	Zero or one (optional)

The example below shows the encoding of faceted search results compliant with [\[OGC17-047\]](#) and includes a “facetedResults” JSON object. The example shows the possible values for the facet “eo:platform”. The OASIS facetedResults XML schema and the JSON Schema in [Annex D.2](#) of the current document allow removing several optional elements to simplify the representation depicted below to obtain a more compact notation.

FacetedResults encoding example

```
{
  "datasource": [
    {
      "displayLabel": "FedEO",
      "description": "Federated Earth Observation",
      "baseURL": "http://fedeo.esa.int/opensearch",
      "facets": [
        {
          "displayLabel": "Platform",
          "index": "eo:platform",
          "relation": "=",
          "terms": [
            {
              "actualTerm": "PROBA-V",
              "query": {
                "count": 1,
                "searchTerms": "vegetation",
                "startIndex": 1,
                "sru:recordSchema": "server-choice",
                "eo:platform": "PROBA-V"
              },
              "requestUrl":
              "http://fedeo.esa.int/opensearch/request?query=vegetation&startRecord=1&maximumRecords=1&httpAccept=application/json&&platform=PROBA-V",
              "count": 24
            },
            {
              "actualTerm": "Aqua",
              "query": {
                "count": 1,
                "searchTerms": "vegetation",
                "startIndex": 1,
                "sru:recordSchema": "server-choice",
                "eo:platform": "Aqua"
              },
              "requestUrl":
              "http://fedeo.esa.int/opensearch/request?query=vegetation&startRecord=1&maximumRecords=1&httpAccept=application/json&&platform=Aqua",
              "count": 1
            },
            {
              "actualTerm": "ENVISAT",
              "query": {
                "count": 1,
                "searchTerms": "vegetation",
                "startIndex": 1,
                "sru:recordSchema": "server-choice",
                "eo:platform": "ENVISAT"
              },
              "requestUrl":

```

```

"http://fedeo.esa.int/opensearch/request?query=vegetation&startRecord=1&maximumRecords
=1&httpAccept=application/json&&platform=ENVISAT",
        "count": 1
    }
]
}
]
}
}

```

DataSource

The following figure shows the DataSource schema.

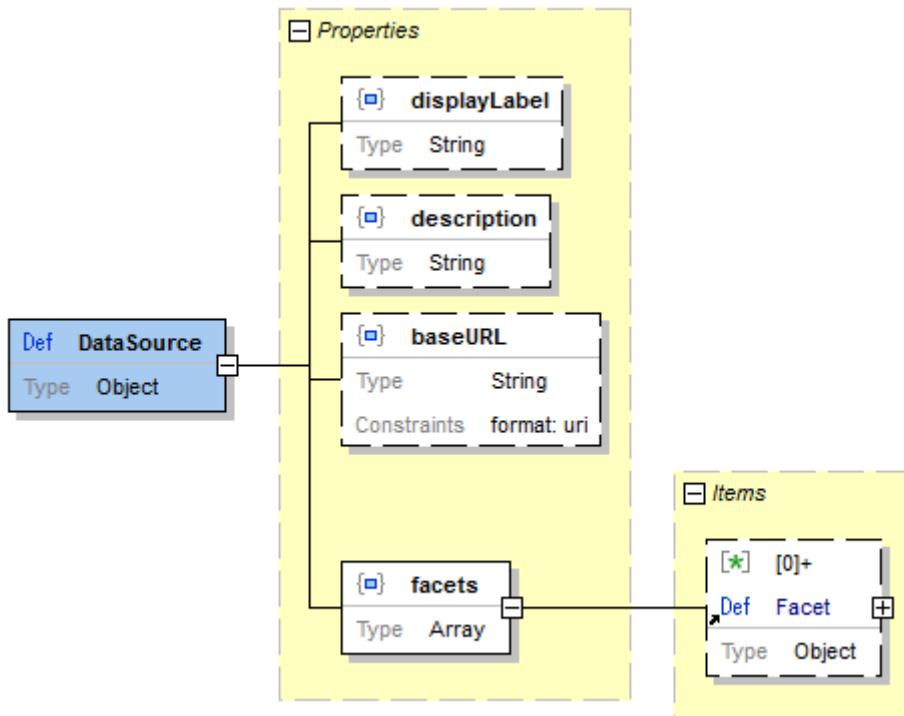


Figure 40. DataSource Schema

The complete description of DataSource is given in [Table 31](#).

Table 31. DataSource object properties

JSON Property	Definition	Data type and values	Multiplicity and use
type \$.datasource.type	Type of the object. This property has the fixed value "DataSource".	Range: String Fixed value: "DataSource"	Zero or one (optional)
displayLabel \$.datasource.displayLabel	Display label for data source.	Domain: DataSource Range: string	Zero or one (optional)

JSON Property	Definition	Data type and values	Multiplicity and use
description \$..datasource.description	Description of data source.	Domain: DataSource Range: string	Zero or one (optional)
baseURL \$..datasource.baseURL	URL representing data source.	Domain: DataSource Range: string	Zero or one (optional)
facets \$..datasource.facets	List of facets and corresponding faceted search results.	Domain: DataSource Range: array of Facet as defined in Table 32 .	Zero or one (optional)

DataSource encoding example

```
{
  "displayLabel": "FedEO",
  "description": "Federated Earth Observation",
  "baseURL": "http://fedeo.esa.int/opensearch",
  "facets": []
}
```

Facet

The following figure shows the Facet schema.

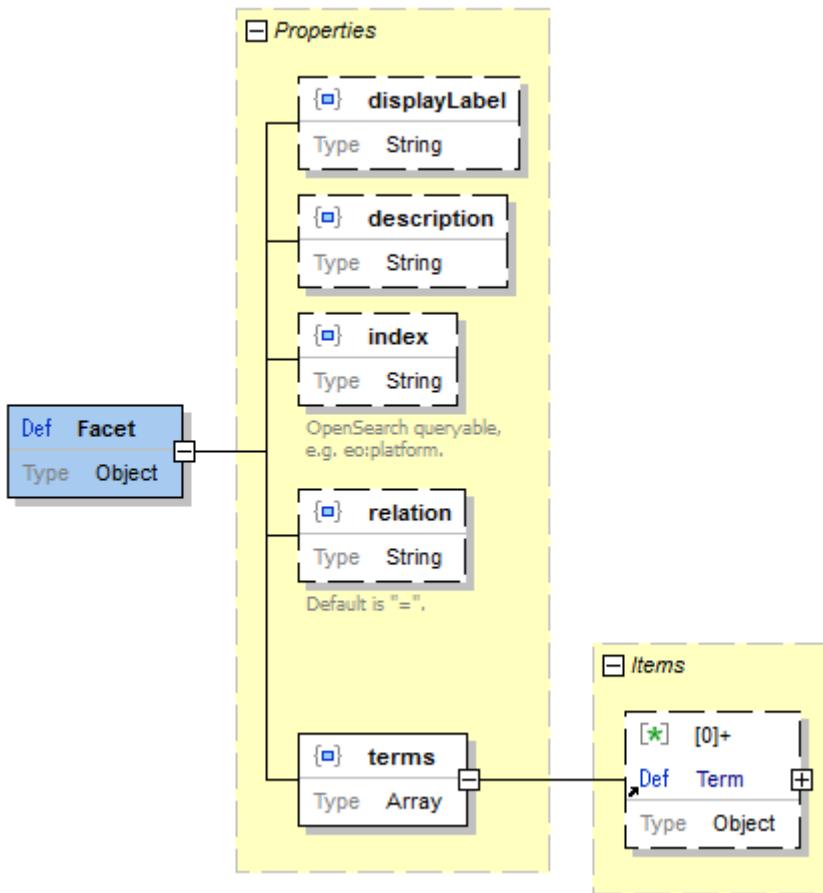


Figure 41. Facet Schema

The complete description of Facet is given in [Table 32](#).

Table 32. Facet object properties

JSON Property	Definition	Data type and values	Multiplicity and use
type \$.facets[*].type	Type of the object. This property has the fixed value "Facet".	Range: String Fixed value: "Facet"	Zero or one (optional)
displayLabel \$.facets[*].displayLabel	Display label for OpenSearch parameter used as an index for the facet count.	Domain: Facet Range: string	Zero or one (optional)
description \$.facets[*].description	Description of the facet.	Domain: Facet Range: string	Zero or one (optional)
index \$.facets[*].index	OpenSearch parameter used as an index for the facet count.	Domain: Facet Range: string	Zero or one (optional)
relation \$.facets[*].relation	Relation used to group search results.	Domain: Facet Range: string	Zero or one (optional)

JSON Property	Definition	Data type and values	Multiplicity and use
terms \$..datasource.terms	List of terms for current facet.	Domain: Facet Range: array of Term as defined in Table 33 .	One (mandatory)

Facet encoding example

```
{
  "displayLabel": "Platform",
  "index": "eo:platform",
  "relation": "=",
  "terms": [
    {
      "actualTerm": "PROBA-V",
      "query": {
        "count": "1",
        "searchTerms": "vegetation",
        "startIndex": "1",
        "srq:recordSchema": "server-choice",
        "eo:platform": "PROBA-V"
      },
      "requestUrl": "http://fedeo.esa.int/opensearch/request?query=vegetation&startRecord=1&maximumRecords=1&httpAccept=application/json&platform=PROBA-V",
      "count": 24
    }
  ]
}
```

Term

The following figure shows the Term schema.

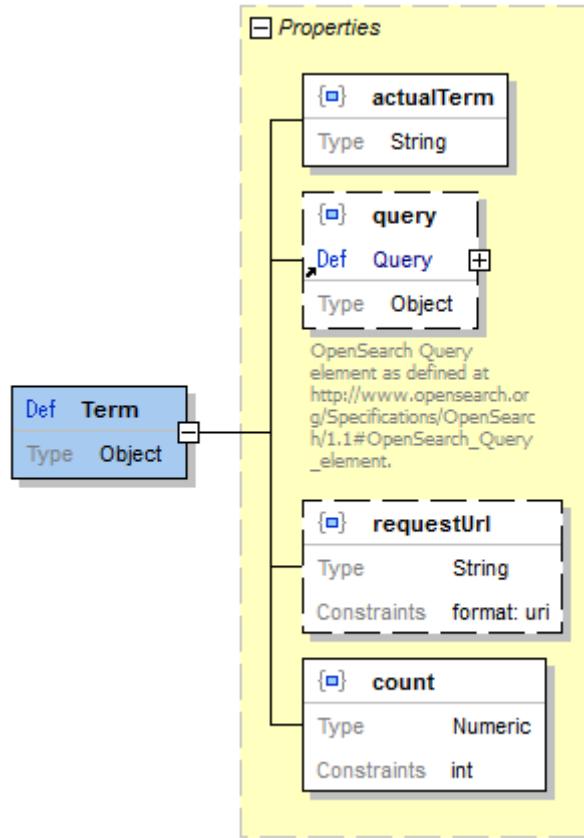


Figure 42. Term Schema

The complete description of Term is given in [Table 33](#).

Table 33. Term object properties

JSON Property	Definition	Data type and values	Multiplicity and use
type \$.terms[*].type	Type of the object. This property has the fixed value "Term".	Range: String Fixed value: "Term"	Zero or one (optional)
actualTerm \$.terms[*].actualTerm	Actual term.	Domain: Term Range: string	One (mandatory)
query \$.terms[*].query	Request parameters to retrieve the search results corresponding to the current term.	Domain: Term Range: Query as defined in OGC 17-047 .	Zero or one (optional)
requestURL \$.terms[*].requestURL	Request URL to retrieve the search results corresponding to the current term.	Domain: Term Range: string (URI)	Zero or one (optional)
count \$.terms[*].count	Number of results corresponding to this term.	Domain: Term Range: integer	One (mandatory)

Term encoding example

```
{  
    "actualTerm": "ENVISAT",  
    "query": {  
        "count": "1",  
        "searchTerms": "vegetation",  
        "startIndex": "1",  
        "srush:recordSchema": "server-choice",  
        "eo:platform": "ENVISAT"  
    },  
    "requestUrl":  
        "http://fedeo.esa.int/opensearch/request?query=vegetation&startRecord=1&maximumRecords  
        =1&httpAccept=application/json&&platform=ENVISAT",  
        "count": 1  
}
```

Chapter 8. Software Design

8.1. Software static architecture

The diagram below shows the original allocation of components to the different participants.

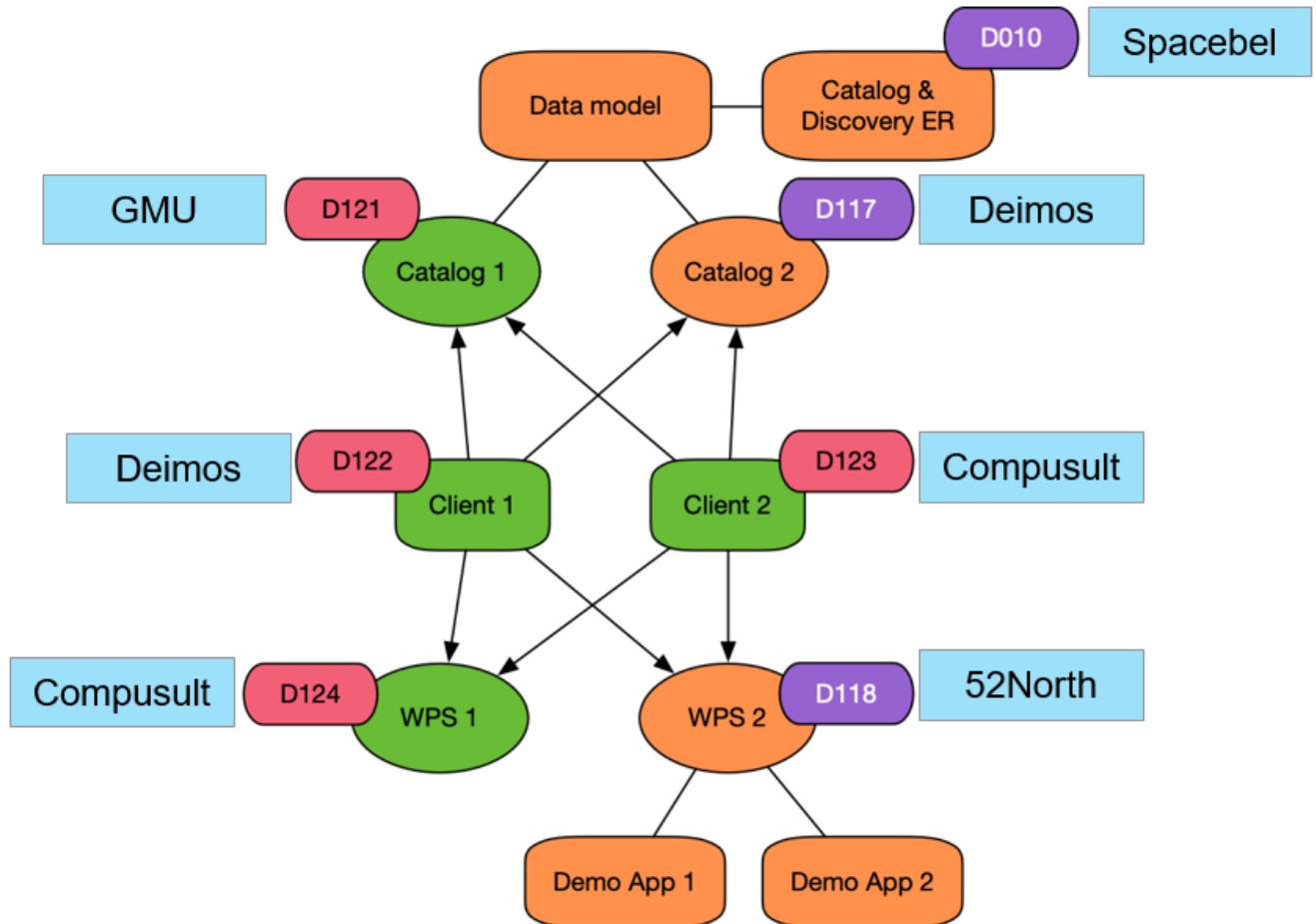


Figure 43. Software static architecture

During the Testbed-15 EOPAD thread, the following client and server-side components were deployed and used for integration testing.

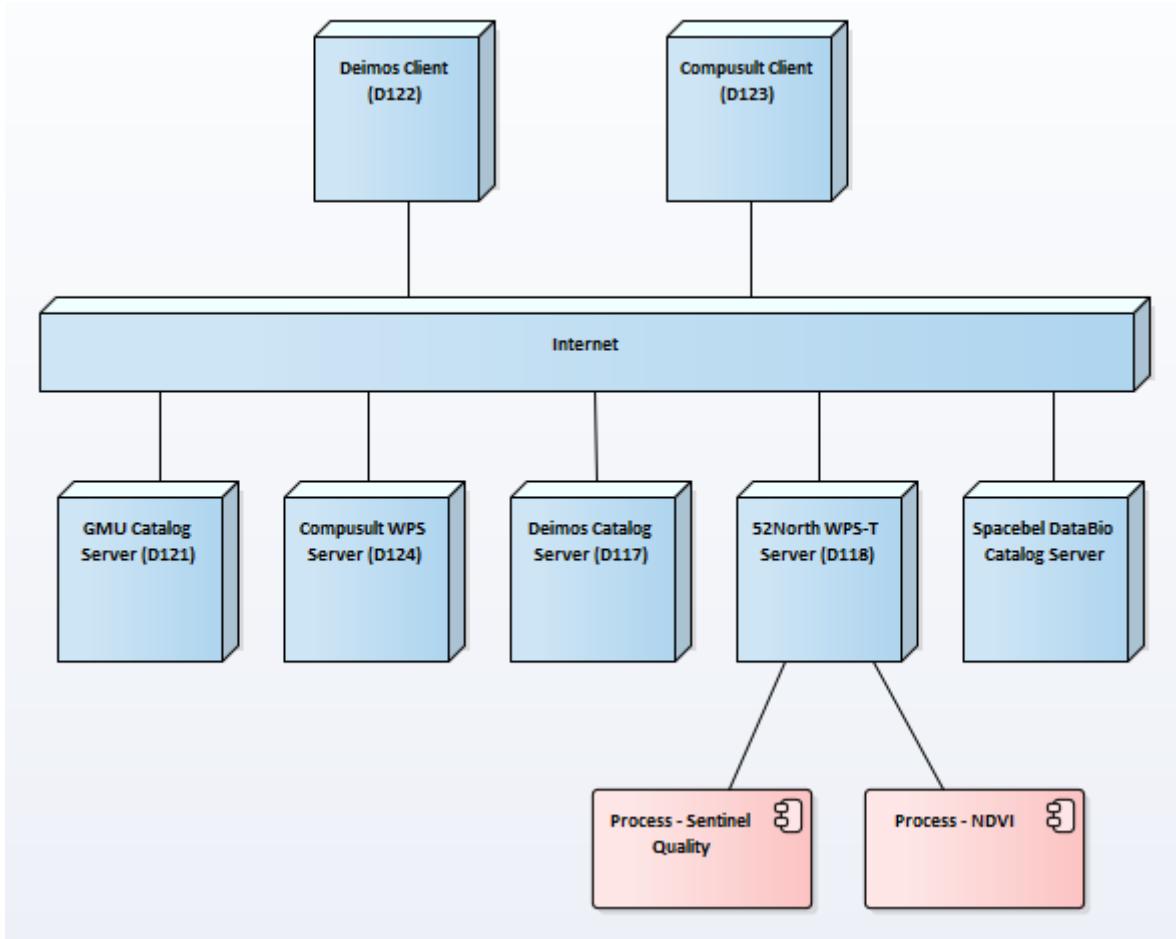


Figure 44. Deployment Diagram

8.2. Interfaces context

The diagram below shows the interoperability tests that were performed.

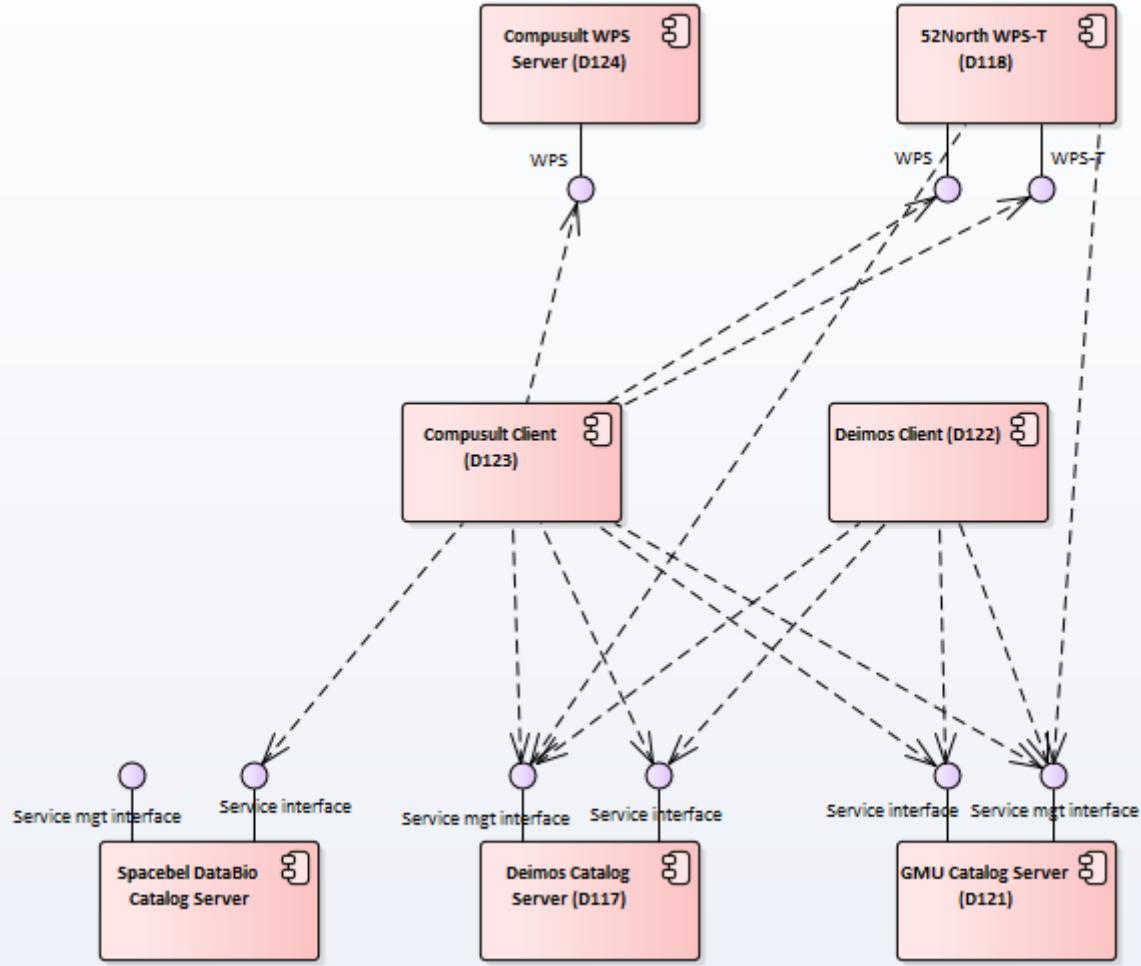


Figure 45. Component Diagram

8.3. Software components design

8.3.1. D117 - Catalog Server (Deimos)

The Deimos software was developed as a plugin for CKAN, an Open Source data portal used in numerous Open Data portals worldwide, including in the Earth Observation domain such as H2020 NextGEOSS or AmeriGEOSS.

The catalogue server provides two interfaces for the users:

Service Interface: This interface allows other catalogue client or another application to discover applications and services through faceted search. This interface has been implemented as an API following the OpenSearch Specification defined in section 6.2 of the Data Model. The python shell of the OpenAPI was generated using the .yaml file and the [Swagger Editor Tool](https://swagger.io/tools/swagger-editor/) [<https://swagger.io/tools/swagger-editor/>]. The catalogue also provides a web based user interface (See "["D122 - Client \(Deimos\)"](#)") to discover new services in the catalogue.

The user is able to discover services via the OpenSearch API as well. Below are some examples of the OpenSearch Query and respective response. In this example the user is querying by organization, bounding box, free text, unique id, keyword and time interval.

[http://servicecatalogue-
ogctestbed15.deimos.pt-smi/services?maximumRecords=2&startRecord=1&organisation_name=ESA/ESRIN&bbox=-180,-90,-100,90&query=near&uid=MultiSensorNDVI&subject=NDVI&startDate=2019-06-19T00:00:00&endDate=2019-06-19T20:30:00](http://servicecatalogue-ogctestbed15.deimos.pt-smi/services?maximumRecords=2&startRecord=1&organisation_name=ESA/ESRIN&bbox=-180,-90,-100,90&query=near&uid=MultiSensorNDVI&subject=NDVI&startDate=2019-06-19T00:00:00&endDate=2019-06-19T20:30:00)

```
{  
  "facetedResults": {  
    "facets": [ ]  
  },  
  "features": [...],  
  "id": "http://servicecatalogue-  
ogctestbed15.deimos.pt-smi/services?maximumRecords=2&startRecord=1&organisation_name=E  
SA%2FESRIN&bbox=-180,-90,-  
100,90&query=near&uid=MultiSensorNDVI&subject=NDVI&startDate=2019-06-  
19T00:00:00&endDate=2019-06-19T20:30:00",  
  "itemsPerPage": 2,  
  "queries": {  
    "request": [  
      {  
        "geo:bbox": "-180,-90,-100,90",  
        "count": 2,  
        "time:end": "2019-06-19T20:30:00",  
        "eo:keyword": "NDVI",  
        "eo:organisationName": "ESA/ESRIN",  
        "os:searchTerms": "near",  
        "time:start": "2019-06-19T00:00:00",  
        "startIndex": 1,  
        "geo:uid": "MultiSensorNDVI"  
      }  
    ]  
  },  
  "startIndex": 1,  
  "totalResults": 1,  
  "type": "FeatureCollection"  
}
```

This interface is aligned with the specifications mentioned in the Use Case C (Application discovery) :

Bob, a user of applications, uses the Catalogue Service to find an application to perform a specific processing algorithm such as NDVI. He fills a form with the available search criteria or uses a free text form to specify some key words and then the service returns all the applications which matches the specified criteria. The interaction between Bob and the Catalogue Service is mediated by a client of the service which provides a GUI to ease the user experience of Bob.

This use case is also implemented as part of the activity **D122 - Client (Deimos)**

Service Management Interface: This interface allows catalog clients or other applications to create, update and delete information about applications/services. This interface also provides a web based user interface to perform the same set of operations.

<http://servicecatalogue-ogctestbed15.deimos.pt-smi/> returns the Landing Page.

The below section provides examples for each of the applicable use cases which have been implemented.

Use Case A (Application/Service lifecycle management from user/operator)

Use Case description 1 : Alice activates the Catalogue Service management interface to register her application (passing all information required by the data model).

The registration of a new service is possible via the Catalogue Frontend and the OpenAPI interface. Below are the examples:

Catalogue frontend:

GeoJSON:

```
{
  "id": "http://fedeo.esa.int/services/MultiSensorNDVI",
  "type": "Feature",
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [
          [
            [
              [
                [
                  -100,
                  -90
                ],
                [
                  100,
                  -90
                ],
                [
                  100,
                  90
                ],
                [
                  -100
                ]
              ]
            ]
          ]
        ]
      ]
    ]
  }
}
```

You can use [Markdown formatting here](#)

Add Service

deimos
elecnor group

OGC
Making location count.

OpenAPI: To submit a service into the catalogue via API, the service owner can make a POST request to an [API url](#) [<http://servicecatalogue-ogctestbed15.deimos.pt-smi/services>] where the body is the service GeoJSON metadata as specified in section 7 of the Data Model.

An additional validation step is performed for both the frontend catalogue as well as the API. The validation step verifies if the service is already catalogued, and if so the application will inform the service owner of the existing conflict, so the service owner can take the proper approach to resolve the conflict.

In case of a successful registration of the service into the catalogue, both a code 201 and the message Service or application has been successfully inserted are return to the service owner, as seen in the following figure.

The screenshot shows the deimos catalogue interface. At the top, there is a navigation bar with the deimos logo, 'Services', 'Organizations', and a blue button labeled 'Add a Service'. Below the navigation bar, the URL 'Add Service' is visible. The main content area has a title 'Service to catalogue'. On the left, there is a 'GeoJSON:' input field containing a JSON object representing service metadata. On the right, a modal dialog box displays the message 'Service or application has been successfully inserted' with an 'OK' button. At the bottom of the page, there is a footer with the deimos logo and the OGC logo.

Use Case Description 2: The service checks that Alice is authorized to register the new application and then fulfill her request.

The catalogue frontend requires users to create a profile and register themselves as an authorized user. The users are needed to provide his/her credentials into the Catalogue user interface to register a new service. The below image illustrates the login page of the catalogue:

Forgotten your password?

No problem, use our password recovery form to reset it.

[Forgot your password?](#)

Login

Username:

Password:

Remember me

[Login](#)



The CKAN solution supports a structure of Organizations. These are used to create, manage and publish collections of services. Users can have different roles within an Organization, depending on their level of authorization. The following roles are currently available in the catalogue:

- **Member** - The user is able to view the datasets that belong to the organization.
- **Editor** - The user has permissions within the organization to publish, edit and delete services.
- **Administrator** - Besides the editor capabilities, the user is also able to delete the organization.

The role within the organization is given and managed by the system administrator upon the user registration. In the figures below, the user on the left has Editor permissions within the Compusult organization, and thus is able to manage the dataset (edit / delete). While the user on the right only has member permissions, and thus is only able to view the dataset. This is visible by the Manage button on the service page.

Field	Value
Last Updated	September 17, 2019, 5:55 PM (UTC+01:00)
Created	September 16, 2019, 6:50 PM (UTC+01:00)
Identifier	D124WPST
Kind	http://purl.org/dc/dcmitype/Service
Lang	EN
Service ID	http://iam-l.compusult.net:8888/wes/west/wps-t-ogc15
Updated	2019-09-18T15:15
Version Info	1.0
Version Notes	Created for use in OGC Testbed-15.

The user authentication specified in the OpenAPI API description is performed via CKAN API Key, which contains the user permissions for each of the Organizations to which the user belongs.

Use Case Description 3 : Alice performs a simple query to be presented with a list of only her applications and checks that the application is registered, asking for its details.

The service returns the record of the application

The Catalogue user interface provides the list of services already registered in the catalogue. To retrieve the list of only the user applications, the user can filter using the Organizations to which they belong.

The screenshot shows a web-based catalogue interface. At the top, there is a navigation bar with the deimos elecnor group logo, a 'Services' button (which is highlighted in blue), an 'Organizations' link, and a 'Add a Service' link. Below the navigation, the page title is 'Datasets'. On the left, there is a sidebar with filters for 'Organizations' (set to 'Compusult'), 'Groups' (empty), 'Tags' (empty), 'Formats' (set to 'JSON'), and 'Licenses' (empty). The main content area has a search bar with the placeholder 'Search datasets...' and a magnifying glass icon. Below the search bar, it says '1 dataset found' and 'Organizations: Compusult'. A single result is listed: 'OGC Testbed 15 WPS-T Service' with the description 'WPS-T service for OGC Testbed 15.' and a 'JSON' button. At the bottom of the content area, it says 'You can also access this registry using the API.'

An example Query to discover the user services via an OpenAPI description could be as follows:
http://servicecatalogue-ogctestbed15.deimos.pt/smi/services?organisation_name=ESA/ESRIN

Use Case Description 4: Alice updates the metadata information of her application
The service checks that Alice is authorized to update the information of the application and then modifies it.

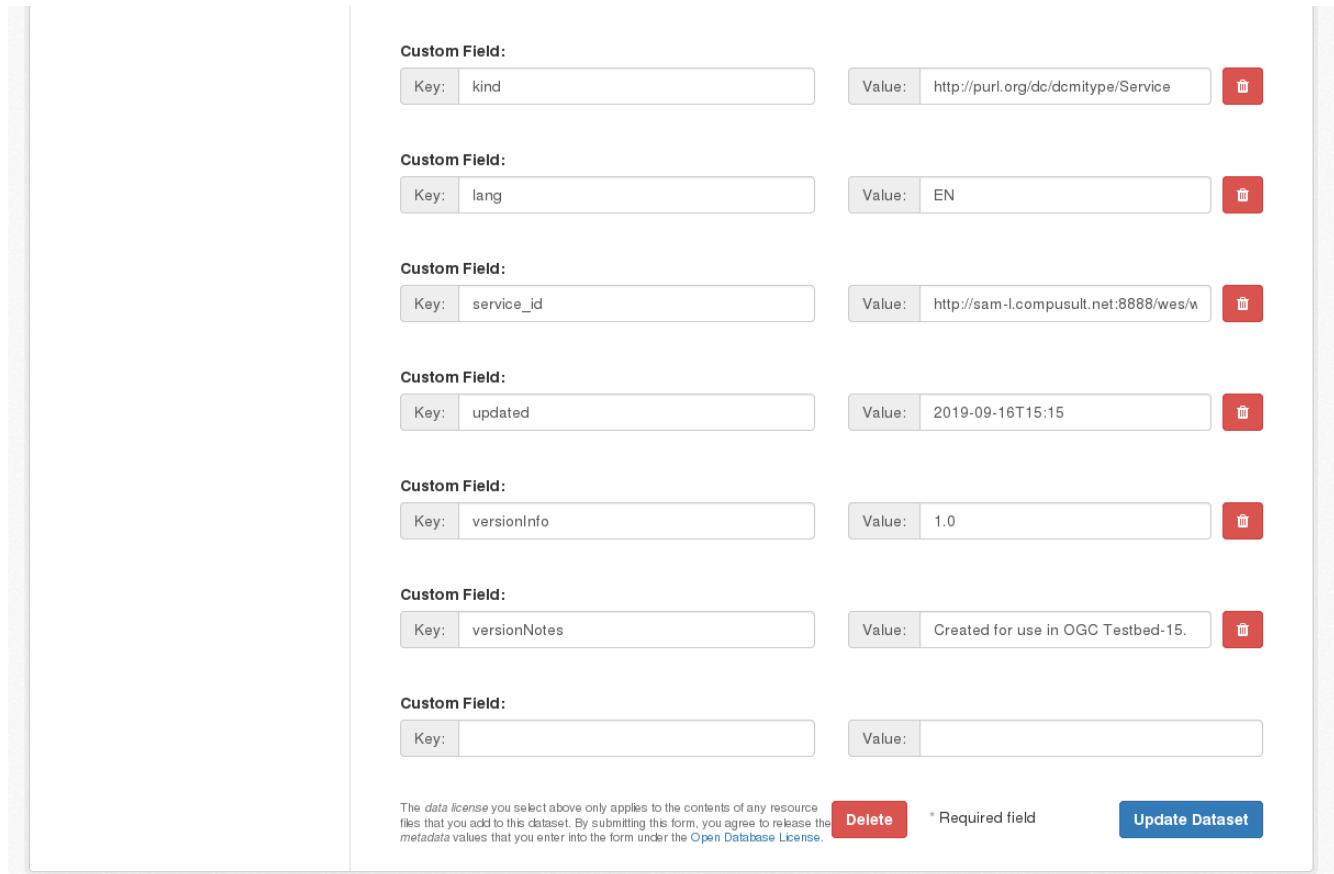
The user is able to update the metadata record of the service via the catalogue user interface and the OpenAPI description. The catalogue ensures that the user is authorized to update the service through the process described previously in the service posting.

Use Case Description 5: Alice requests the deletion of the application in the Catalogue Service. The service checks that Alice is authorized to delete the information of the application and then deletes it

The user is able to delete the metadata record of the service via the catalogue user interface and the

OpenAPI description. The catalogue ensures that the user is authorized to delete the service through the process described previously in the service posting.

The user can update and/or delete a service through the service management page in the catalogue front-end (Figure below) or via PUT or DELETE request via OpenAPI description (example provided below). In this case, the user needs to mention the service name in the request.
<http://servicecatalogue-ogctestbed15.deimos.pt-smi/services/multisensorndvi>



The screenshot shows a form for managing service metadata. It consists of several sections, each labeled 'Custom Field:' with two input fields: 'Key' and 'Value'. Each section also includes a red trash can icon for deletion. The fields listed are:

- Custom Field:
Key: kind Value: <http://purl.org/dc/dcmitype/Service>
- Custom Field:
Key: lang Value: EN
- Custom Field:
Key: service_id Value: <http://sam-l.compusult.net:8888/wes/m>
- Custom Field:
Key: updated Value: 2019-09-16T15:15
- Custom Field:
Key: versionInfo Value: 1.0
- Custom Field:
Key: versionNotes Value: Created for use in OGC Testbed-15.
- Custom Field:
Key: (empty) Value: (empty)

At the bottom of the form, there is a note: 'The data license you select above only applies to the contents of any resource files that you add to this dataset. By submitting this form, you agree to release the metadata values that you enter into the form under the [Open Database License](#)'. Below this note are three buttons: a red 'Delete' button, a red 'Required field' note, and a blue 'Update Dataset' button.



Use Case B (Application creation/deletion via M2M interaction)

The catalogue implementation is aligned with the Use case B where the following set of operations can be performed with the OpenAPI interface provided by the catalogue. All the features have been explained in the previous Use Case A.

Metadata published

Using the CKAN solution there are mandatory metadata fields when publishing a service, such as identifier and title, while the remaining metadata fields are stored as extra fields at the discretion of the catalogue owner. Also using a feature of CKAN the entire GeoJSON is stored as a resource of the service. The figure below illustrates some of the metadata fields catalogued and displayed to the users.

Multi Sensor NDVI

Followers 0

+ Follow

Social

- Twitter
- Facebook

Multi Sensor NDVI

NDVI is calculated after the two bands values Near Infrared and red. It is calculated by this formula : NDVI = (NIR-Red)/(NIR+Red)

Data and Resources

multisensordv.json
Service GeoJSON file

Explore ▾

ESA OADS Earth Online NDVI

Additional Info

Field	Value
State	active
Last Updated	September 18, 2019, 11:42 AM (UTC+01:00)
Created	September 18, 2019, 11:42 AM (UTC+01:00)
Organization	ESA/ESRIN
endpointURL	http://tep.esa.int/wps/processing?REQUEST=GetCapabilities&SERVICE=WPS
identifier	MultiSensorNDVI
kind	http://inspire.ec.europa.eu/metadata-codelist/ResourceType/service
lang	en
service_id	http://fedeo.esa.int/services/MultiSensorNDVI
spatial	{"type": "Polygon", "coordinates": [[[-100, -90], [100, -90], [100, 90], [-100, 90], [-100, -90]]]}
updated	2019-06-19T00:00:00Z
versionInfo	3.0
versionNotes	Updated for use in OGC Testbed-15.

Interoperability tests

An interoperability test was carried out between this catalogue server (D117) and the clients (D122 and D123). The goal of this test is to validate the ability of the clients to register a new service to the catalogue and browse, discover update and delete the services using the API exposed by the catalogue. The results of these tests are presented below:

deimos
deimos
Organizations 52 North Processing API at ...

Followers 0

+ Follow

Organization

52 North
There is no description for this organization

Social

- Twitter
- Facebook
- License

Creative Commons Attribution

Processing API at testbed.dev.52north.org

Data and Resources
[org.52north.dev.testbed.depotdev.json](#)
Service GeoJSON file

Explore ▾

Additional Info

Field	Value
State	active
Last Updated	September 23, 2019, 15:10 AM (UTC+01:00)
Created	September 23, 2019, 15:09 AM (UTC+01:00)
identifier	org.52north.dev.testbed.devtest
kind	http://inspire.ec.europa.eu/metadata-codelist/ResourceType/service
service_id	org.52north.dev.testbed.devtest
updated	2019-09-23T15:09:52.588Z

deimos
deimos
Organizations Compusult OGC Testbed 15 WPS-T Service

Followers 0

+ Follow

Organization

Compusult
There is no description for this organization

Social

- Twitter
- Facebook
- License

Creative Commons Attribution

OGC Testbed 15 WPS-T Service

WPS-T service for OGC Testbed 15.

Data and Resources
[d124wpt.json](#)
Service GeoJSON file

Explore ▾

Additional Info

Field	Value
State	active
Last Updated	September 17, 2019, 5:55 PM (UTC+01:00)
Created	September 16, 2019, 6:50 PM (UTC+01:00)
identifier	D124WPT
kind	http://purl.org/ogc/dcm/p/Service
lang	EN
service_id	http://nam1.compusult.net:8888/www/wps1/wpt-reg15
updated	2019-09-16T18:15
versionInfo	1.0
versionNotes	Created for use in OGC Testbed-15.

Lessons Learned

During the development of the catalogue and respective OpenAPI, it has been found that there are some topics that require further discussion in order to determine the correct path ahead.

Catalogue:

The parsing of the service GeoJSON can quickly become quite complex and extensive due to the fact that there is a high variance and length that can occur within the structure of any given GeoJSON feature. In the solution adopted (CKAN), the complete GeoJSON is stored as a resource, and only a few metadata fields are collected. If this solution is adopted, it is recommended that a metadata field list is compiled, such that it contains the fields that are queryable. As an example, the field that contains the default value of certain input maybe holds no interest querywise, and thus would not be catalogued, but this value would still be within the complete service GeoJSON (resource).

The catalogue can be extended to categorise the services based on the offering types for example - WPS, WMS, Application Package etc.

OpenAPI:

During the development OpenAPI there were two topics that will require further discussion in further iterations.

The first topic regards the OpenSearch Description Document, where the Parameter Extension of OpenSearch is not expressive enough to be able to represent the different components inside a single parameter value. Such parameter is sru:facetLimit, which can assume different values and structures: fixed strings (allowed facets), fixed integers (no facet search), integers set by the user or a mix of fixed strings and dynamic integer.

The second topic regards the authorization / security permission within both the OpenAPI and Catalogue. Since there were no requirements / guidelines regarding the authentication and authorization, the approach adopted was to use the user manager incorporated into our catalogue solution, and propagate it to the OpenAPI via API-KEY header. However, if this topic is not thoroughly discussed, it might lead to multiple implementations by each developer. Leading to the clients to adapt to each API instead of having a uniform way of access.

8.3.2. D121 - Catalog Server (GMU)

Description

The implementation of the catalogue service is implemented and integrated with GeoNetwork – an Open Source catalog application. The basic design is to internally map input or output in GeoJSON to metadata in ISO 19139. This design and development resulted in re-using the browsing capabilities of GeoNetwork without much changes.

Interfaces implemented

The interfaces implemented in GMU Catalog service are shown in [Table 34](#).

Table 34. Interfaces and Specifications Implemented by GMU OGC Catalog Service

Services	URL	Specifications
OpenAPI service	https://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a	OpenAPI + GeoJson EO. Refer to Catalogue Service Specification for details. Related document numbers: OGC 17-047r1, OGC 07-045r1, OGC 17-003r0
Browse Page	Catalog Browse [https://cloud.csiss.gmu.edu/geonet/]	Geonetwork-based view & browse
CAT 3.0	https://cloud.csiss.gmu.edu/ows15/geonet/srv/eng/cat3	Related document numbers: OGC 12-168r6, OGC 12-176r7
CSW 2.0.2	https://cloud.csiss.gmu.edu/ows15/geonet/srv/eng/csw	Related document numbers: OGC 07-06r1, OGC 13-084r2

Metadata published

The GMU Catalog service is provided as a transactional service that allows the insertion/editing/deletion of records that can be injected by either of the following media types:

- Requests and responses in GeoJSON EO encoding as specified in [\[Chapter-6\]](#).
- Requests and responses in ISO 19139 encoding through CAT 3.0
- Requests and responses in ISO 19139 encoding through CSW 2.0.2

For this testbed, the OpenAPI is a new development. The following typical use cases are given and exemplified in details.

OpenAPI

The OpenAPI is implemented as specified in [\[Chapter-6\]](#). The API document is posted in SwaggerHub where code-generation of server-stub and client-stub can be automated. The interface can be viewed in [GMU Catalog Service OpenAPI interface](#).

The screenshot shows the GMU Catalog Service OpenAPI interface on SwaggerHub. The main title is "Testbed 15 Service Catalogue" with version "1.0.0 OAS3". Below the title, there is a note: "Test Service, following ISO/OGC interface guidelines, to Earth Observation metadata." and links to "CSISS - Website", "Send email to CSISS", "Apache 2.0", and "Landing page for the API".

Servers: https://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a

Describe: Describe the catalogue interface.

- GET /** provide landing page (rel="self") in OAPI Common format.
- GET /conformance** provide conformance class information (rel="conformance") in API Common format.
- GET /api** describe the catalogue interface (rel="service")

Search: Search for service or application.

- GET /services** service or application search (rel="data")
- GET /services/{service-id}**

Insert: Insert new service or application.

- POST /services**

Update: Update existing service or application.

- PUT /services/{service-id}**

Delete: Delete a service or application.

- DELETE /services/{service-id}**

Figure 46. GMU Catalog Service OpenAPI interface

Use Case 1 - Landing Page

The first use case is to show what is returned by accessing the landing page. One example request is shown in Table 35.

Table 35. Use Case 1 - Landing Page of GMU OGC Catalog Service

Item	Content	
Path	https://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a	
Request	Method	GET
	Headers	Accept: application/json
Full request example by Curl	<pre>curl --header "Accept: application/json" https://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3</pre>	

Item	Content	
Response	Status	HTTP/1.1 200
	Headers	Content-Type: application/json; charset=UTF-8
	Body	<pre>{"links": [{"href": "http://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a/", "rel": "self", "type": "application/json", "title": "OpenAPI Catalog Landing Page"}, {"href": "http://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a/api", "rel": "service", "type": "application/openapi+json;version=3.0", "title": "OpenAPI Definition in JSON"}, {"href": "http://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a/api", "rel": "service", "type": "text/yaml", "title": "OpenAPI Definition in YAML"}, {"href": "http://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a/api", "rel": "service", "type": "application/sru+xml", "title": "OpenAPI Definition in SRU"}, {"href": "http://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a/conformance", "rel": "conformance", "type": "application/json", "title": "OGC conformance class implemented by this Catalog"}, {"href": "http://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a/services", "rel": "data", "type": "application/json", "title": "Services and application metadata"}]}</pre>

Use Case 2 - Get OpenAPI specification

The specification of OpenAPI can be retrieved in different formats. One request to retrieve specification in YAML is shown in [Table 36](#).

Table 36. Use Case 2 - Get OpenAPI specification

Item	Content	
Path	https://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a/api	
Request	Method	GET
	Headers	Accept: text/yaml
Full request example by Curl	<pre>curl --header "Accept: text/yaml" https://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a/api</pre>	

Item	Content	
Response	Status	HTTP/1.1 200
	Headers	Content-Type: text/yaml;charset=UTF-8
	Body	<pre> openapi: "3.0.0" info: description: "Test Service, following ISO/OGC interface guidelines, to Earth Observation\ \ metadata." title: "Testbed 15 Service Catalogue" contact: name: "CSISS" url: "http://csiss.gmu.edu" email: "gyu@gmu.edu" version: "1.0.0" license: name: "Apache 2.0" url: "http://www.apache.org/licenses/LICENSE-2.0.html" x-logo: url: "http://csiss.gmu.edu/csiss_logo_big.gif" x-providerName: "Copyright 2019, George Mason University." x-serviceName: "OGCTestCatalogue" servers: - url: "https://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a " </pre>

Use Case 3 - Get Service Conformance

The example in Table 37 shows the request to retrieve service conformance declaration.

Table 37. Use Case 3 - Get Service Conformance

Item	Content	
Path	https://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a/conformance	
Request	Method	GET
	Headers	Accept: application/json
Full request example by Curl	<pre> curl --header "Accept: application/json" https://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3/conformance </pre>	

Item	Content	
Response	Status	HTTP/1.1 200
	Headers	Content-Type: application/json; charset=UTF-8
	Body	<pre>{"conformsTo": ["http://www.opengis.net/spec/owc-geojson/1.0/req/core", "http://www.opengis.net/spec/os-geojson/1.0/req/core", "http://www.opengis.net/spec/eopad-geojson/1.0/req/core", "http://www.opengis.net/spec/OAPI_Common/1.0/req/oas30", "http://www.opengis.net/spec/OAPI_Communication/1.0/req/geojson"]}</pre>

Use Case 4 - Get All Services

The path "/services" provide the base to search and retrieve services by given criteria. All the services can be retrieved if no parameter is used. [Table 38](#) shows the request and its skeleton response.

Table 38. Use Case 4 - Get All Services

Item	Content	
Path	https://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a/services	
Request	Method	GET
	Headers	Accept: application/json
Full request example by Curl	<pre>curl --header "Accept: application/geo+json" curl --header "Accept: application/geo+json" https://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a/services</pre>	
Response	Status	HTTP/1.1 200
	Headers	Content-Type: application/geo+json
	Body	<pre>{"@context": "https://www.opengis.net/spec/os-geojson/1.0", "type": "FeatureCollection", "id": "https://cloud.csiss.gmu.edu/ows15/rest3a/ogc/cat3a/services", "properties": {}, "features": [.....,], "totalResults": 10, "startIndex": 1, "itemsPerPage": 10, "queries": {"type": "Queries", "request": [{"type": "Query"}]}}</pre>

Use Case 5 - Browse All Services

When services are registered into the catalog, they can be browsed using the browser of GeoNetwork. The portal is given in [Table 39](#). The viewer is exemplified in [Use Case 5 - Browse GMU Catalog Services](#).

Table 39. Use Case 5 - Browse All Services

Item	Content
Path	https://cloud.csiss.gmu.edu/geonet/

The screenshot shows a web browser window with the URL https://cloud.csiss.gmu.edu/geonet/srv/eng/catalog.search#/search?facet.q=type%2Fservice. The page displays a search interface with a search bar and a map view on the right. On the left, there is a filter sidebar with options for 'TYPE OF RESOURCES' (Service) and 'PROVIDED BY' (Testbed 15 Experiment Catalog). The main content area shows three service entries with star ratings and small icons. A message at the top indicates cookie acceptance.

Figure 47. Use Case 5 - Browse GMU Catalog Services

Use Case 6 - Transaction - insert

An example insert transaction request is shown in Table 40.

Table 40. Use Case 6 - GMU Catalog Service Transaction - Insert

Item	Content				
Path	https://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a/services				
Request	<table border="1"> <tr> <td>Method</td> <td>POST</td> </tr> <tr> <td>Headers</td> <td> Accept: application/json Content-Type: application/geo+json </td> </tr> </table>	Method	POST	Headers	Accept: application/json Content-Type: application/geo+json
Method	POST				
Headers	Accept: application/json Content-Type: application/geo+json				
Full request example by Curl	<pre>curl -u yourusername:yourpassword -d "@data.json" -X POST --header "Accept: application/geo+json" --header "Content-Type: application/geo+json" https://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a/services</pre>				

Item	Content	
Content of data.json	<pre>{"type": "Feature", "id": "http://52north.org/testbed15/eopad/SentinelQualityAreaOfInterestTEST",}</pre>	
Response	Status	HTTP/1.1 200
	Headers	Content-Type: application/geo+json; charset=UTF-8
	Body	<pre>{"description": "Service or application has been successfully inserted"}</pre>

Use Case 7 - Get a Specific Service by id

The service metadatadata can be retrieved by using a specific identifier in the path as shown in [Table 41](#).

Table 41. Use Case 7 - GMU Catalog Service Get a Specific Service by ID

Item	Content	
Path	https://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a/services/org.52north.dev.testbed.eopad.rest	
Request	Method	GET
	Headers	Accept: application/geo+json
Full request example by Curl	<pre>curl --header "Accept: application/geo+json" -X GET https://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a/services/org.52north.dev.testbed.eopad.rest</pre>	
Response	Status	HTTP/1.1 200
	Headers	Content-Type: application/geo+json
	Body	<pre>{"id": "https://testbed.dev.52north.org/eopad/rest", "type": "Feature", "properties": {"identifier": "org.52north.dev.testbed.eopad.rest", "kind": "http://inspire.ac.europa.eu/metadata-codelist/ResourceType/service", "title": "52°North EOPAD javaPS",}}</pre>

Use Case 8 - Transaction - update a Specific Service

[Table 42](#) shows one example to update a speccific service by using identifier.

Table 42. Use Case 8 - GMU Catalog Service Transaction - update a Specific Service

Item	Content	
Path	https://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a/services/org.52north.dev.testbed.eopad.rest	
Request	Method	PUT
	Headers	Accept: application/json Content-Type: application/geo+json
Full request example by Curl	<pre>curl -u yourusername:yourpassword -d "@data2.json" -X PUT --header "Accept: application/json" --header "Content-Type: application/geo+json" https://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a/services/org.52north.dev.testbed.eopad.rest</pre>	
Content of data.json	<pre>{"id":"https://testbed.dev.52north.org/eopad/rest","type":"Feature","properties":{"identifier":"org.52north.dev.testbed.eopad.rest","kind":"http://inspire.ac.europa.eu/metadata-codelist/ResourceType/service","title":"52°North EOPAD javaPS"}, }</pre>	
Response	Status	HTTP/1.1 200
	Headers	Content-Type: application/json; charset=UTF-8
	Body	<pre>{"description":"Service has been successfully updated."}</pre>

Use Case 9 - Transaction - delete a Specific Service

A specific service can be deleted as shown in Table 43.

Table 43. Use Case 9 - GMU Catalog Service Transaction - delete a Specific Service

Item	Content	
Path	https://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a/services/org.52north.dev.testbed.eopad.rest	
Request	Method	DELETE
	Headers	Accept: application/json
Full request example by Curl	<pre>curl -u yourusername:yourpassword -d "@data2.json" -X DELETE --header "Accept: application/json" https://cloud.csiss.gmu.edu/ows15/geonet/rest3a/ogc/cat3a/services/org.52north.dev.testbed.eopad.rest</pre>	

Item	Content	
Response	Status	HTTP/1.1 204
	Headers	Content-Type: application/json; charset=UTF-8
	Body	(No content returned)

Interoperability tests

The Catalog service OpenAPI has been tested and verified with examples. These examples include example-catalog-insert-ndvi, example-catalog-insert-sentinel-quality, and example-catalog-insert-wps-service. Service discovery and transaction use cases have been tested.

Lessons learned

The GMU catalog service was implemented following the specification outlined in [Catalogue Service Specification](#). The test cases showed that specifications worked in supporting discovery and transaction of services in a fully-functional catalog with GeoJSON encoding schema. The specifications, including the newly released standard – OGC 17-047r1, are comprehensive in fulfillment of service transaction and discovery using a resource-oriented approach.

8.3.3. D122 - Client (Deimos)

Description

The [Catalogue Server D117](#) also provides a web based user interface to discover new services in the catalogue. This service is accessible at <http://servicecatalogue-ogctestbed15.deimos.pt/>.

OGC Testbed 15

Testbed-15 aims to explore new levels of interoperable geospatial data processing with a focus on data-centric security, federated clouds, service & application discovery, portrayal, machine learning, and delta updates.

The Testbed-15 initiative provides a business opportunity for stakeholders to mutually define, refine, and evolve service interfaces and protocols in the context of hands-on experience and feedback. The outcomes are expected to shape the future of geospatial software development and data publication. Testbed Sponsors are supporting this vision with cost-sharing funds to partially offset the costs associated with development, engineering, and demonstration of these outcomes. This offers selected Participants a unique opportunity to recoup a portion of their initiative expenses.

Search data

E.g. environment

Popular tags: NDVI, Earth Online, ESA OADS

Statistics

3 datasets 4 organizations

The catalogue provides a web based user interface to discover services by browsing through the catalogue and/or by searching in a free text search box. The figure below shows the search capability of the catalogue using free text, for example **ndvi** etc.

▼ Organizations

- 52 North 1
- ESA/ESRI 1

▼ Groups

There are no Groups that match this search

▼ Tags

- NDVI 2
- Earth Online 1
- ESA OADS 1
- Sentinel-2 1
- Vegetation 1

▼ Formats

- JSON 2

▼ Licenses

- Creative Commons At... 2

+ Add Dataset

Q

2 datasets found for "ndvi"

Order by: Relevance ▾

Multi Sensor NDVI

NDVI is calculated after the two bands values Near Infrared and red. It is calculated by this formula : $NDVI = (NIR - Red) / (NIR + Red)$

JSON

Calculation of NDVI using the SNAP toolbox for Sentinel-2

Calculation of NDVI using the SNAP toolbox for Sentinel-2

JSON

You can also access this registry using the [API](#).

For this task DEIMOS has also developed a web based client tool to allow users to register, delete and discover services to two catalogues (D117- DEIMOS and D121- GMU). This client tool also allows the users to deploy applications in multiple clouds and compose and execute service chain workflows. The sections below will provide an overview of this client tool:

Register Service

Information for the Registration

Select the catalog to register your service

D117 Deimos
D122 GMU

Attach your service metadata file

Browse

Choose file...

Register

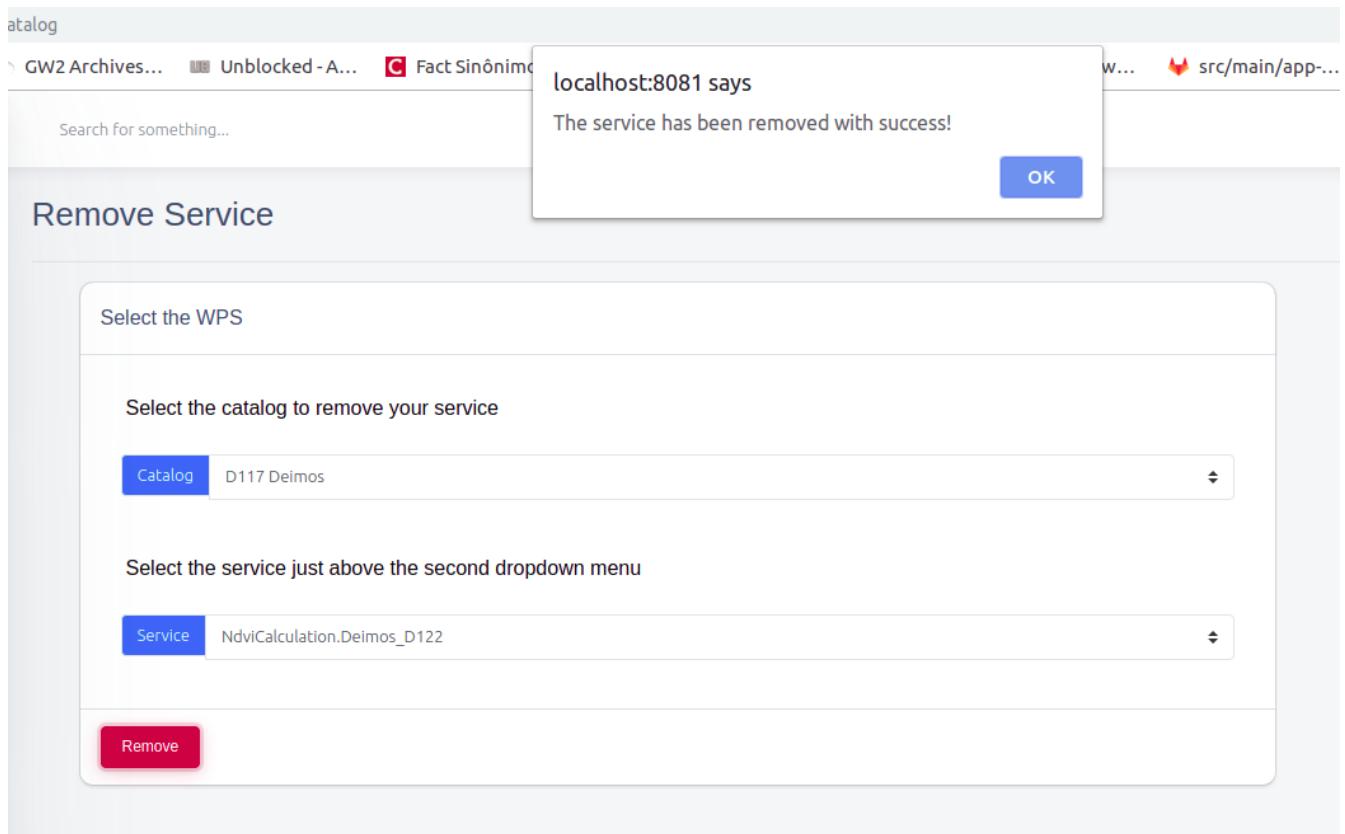
The above image shows the interface where the user can choose the catalogue to register services by providing the GeoJSON file containing the metadata of the service. In this case the user can choose either the Deimos Catalogue (D117) or GMU catalogue (D121) to register the service.

The screenshot shows a web application interface for registering a service. At the top, there are tabs for "GW2 Archives...", "Unblocked - A...", and "Fact Sinônima...". Below the tabs, a search bar contains the placeholder "Search for something...". A modal dialog box is centered on the page, displaying the message "localhost:8081 says" followed by "The service has been registered with success!" and a blue "OK" button. The main form area has a title "Register Service" and a section titled "Information for the Registration". It includes fields for selecting a catalog ("Select the catalog to register your service") which shows "D117 Deimos" in a dropdown, attaching a service metadata file ("Attach your service metadata file") which shows "test.json" in a file input field with a "Browse" button, and a "Register" button at the bottom.

The web client will notify the users once the service has been successfully registered. The above example shows the message when the user registers a service in the DEIMOS Catalogue (D117)

The screenshot shows a web application interface for removing a service. The title is "Remove Service". A modal dialog box is centered on the page, displaying the message "Select the WPS" and "Select the catalog to remove your service". It includes a dropdown menu for "Catalog" with options "D117 Deimos" and "D122 GMU", a dropdown menu for "Service" with a single item "Select the service just above the second dropdown menu", and a red "Remove" button at the bottom.

Similarly, the user can delete an already registered service from the Deimos Catalogue (D117) or GMU catalogue (D121). The users are presented with the list of services registered in the catalogue to choose the service to delete. The user access mechanism ensures that the user has the sufficient access privileges to delete the service.

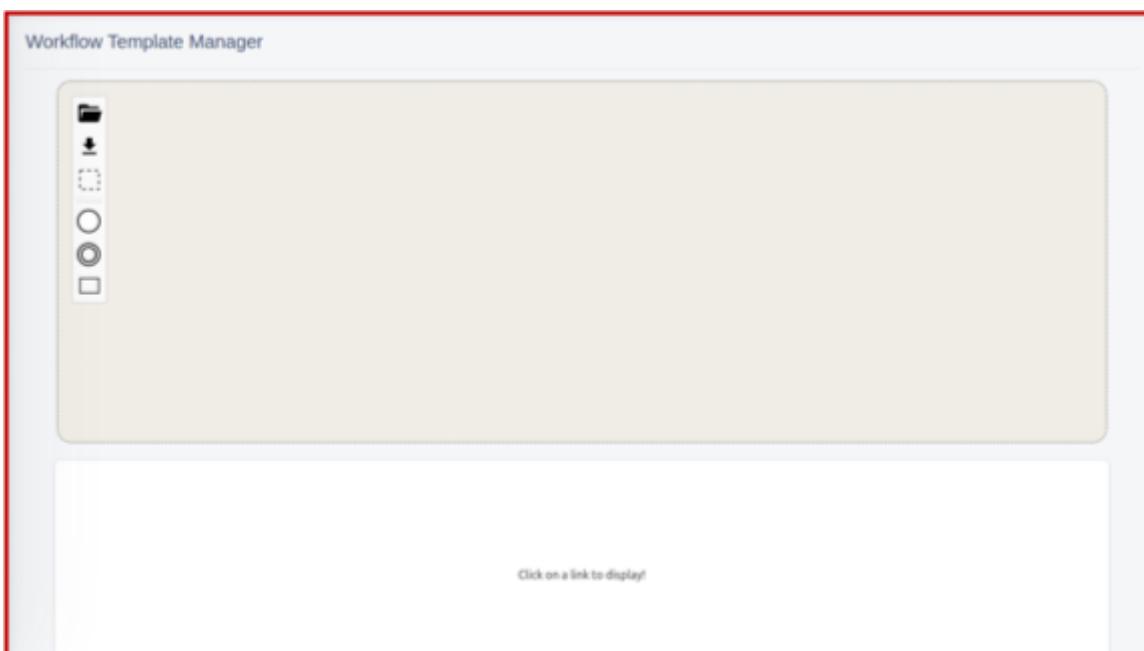


The users are presented with a message once the service has been removed successfully from the catalogue.

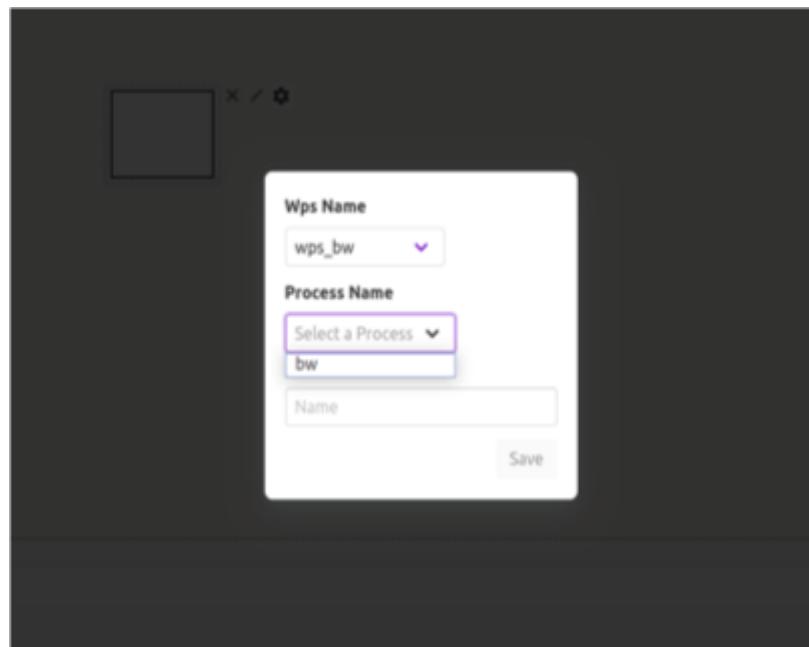
The users can discover services registered in both the Catalog : DEIMOS and GMU from this client. The below image shows the service discovery interface:

The results of the service discovery is presented in the client user interface. The below image illustrates that:

The users can compose a workflow using this web based tool. The below example shows the workflow composition canvas.



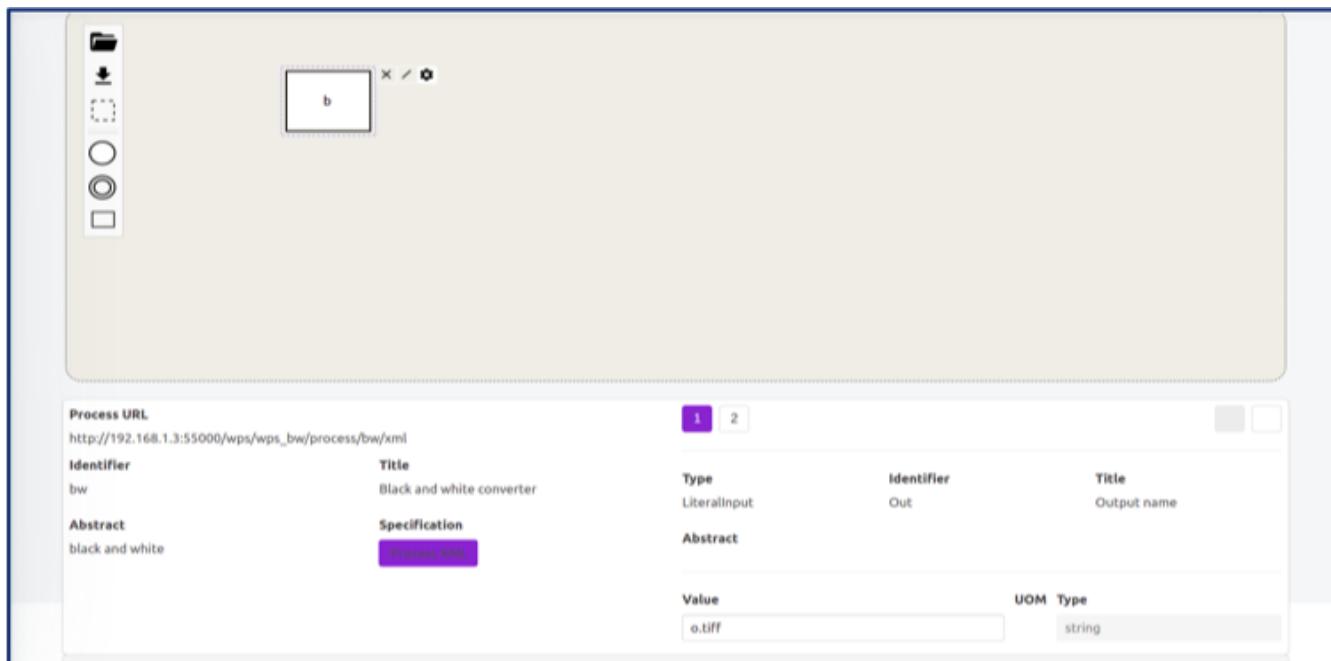
The users can drag and drop the components and configure the components to identify the WPS.



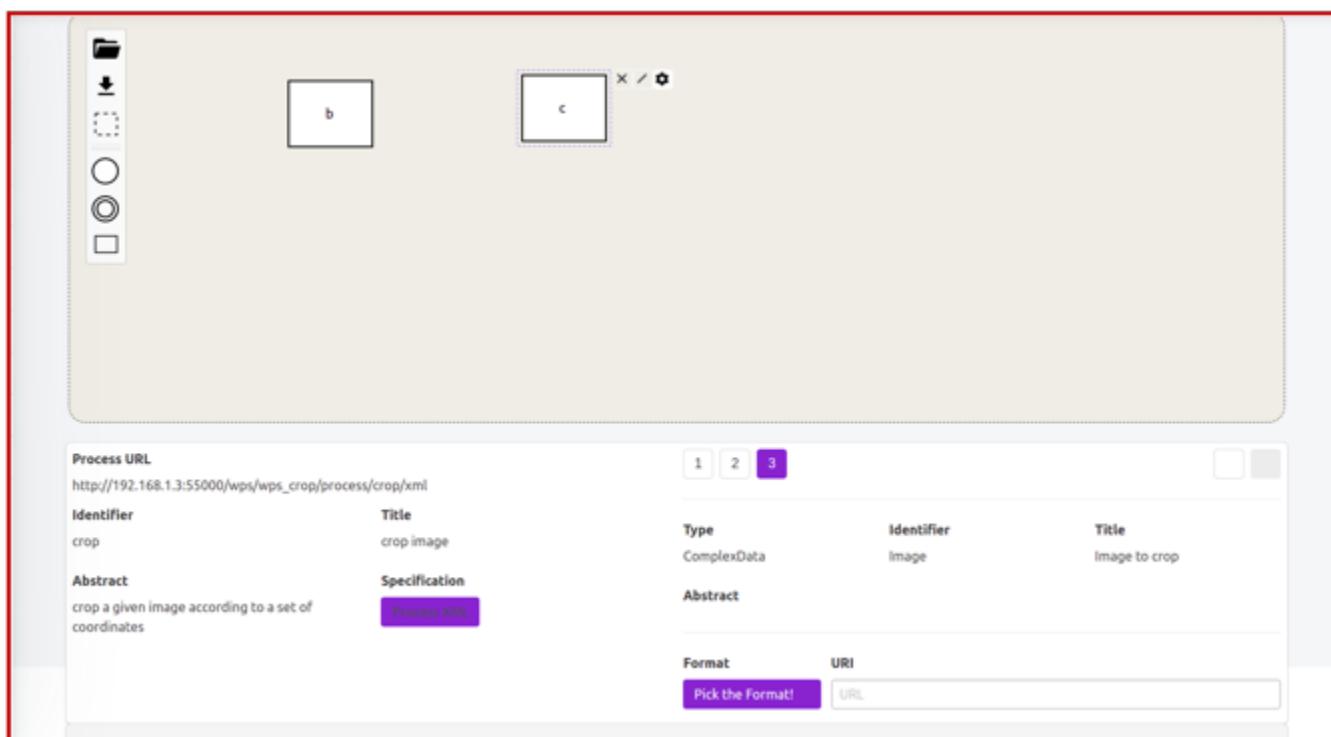
In this case the user is configuring a processing module by selecting the WPS service. The drop-down menu presents the users with the list of WPS services discovered from the catalogue. The tool validates the WPS against the OGC standards before processing further.

A screenshot of a software interface for managing processing models. On the left, there is a sidebar with icons for folder, download, and other operations. In the center, there is a workspace with a small icon labeled 'b'. Below the workspace, there is a table with two rows of data. The first row contains 'Process URL' (http://192.168.1.3:55000/wps/wps_bw/process/bw/xml), 'Identifier' (bw), 'Title' (Black and white converter), 'Type' (ComplexData), 'Identifier' (Image), and 'Title' (image to black and white). The second row contains 'Abstract' (black and white), 'Specification' (Process XML), 'Abstract' (Abstract), 'Format' (0: image/png), and 'URI' (https://cdn.pixabay.com/photo/2016/02/04/13/49/the-earth-1179212_960).

Once the processing model is validated, the user needs to configure the input parameter. In the above example the reference to the input data is provided along with the type of data.



Once the processing model is validated , the user needs to configure the output parameter. The user defines the output data identifier.



the above example shows inclusion of the following service in the service chain and configuring the service in the similar way described earlier.

Process URL: http://192.168.1.3:55000/wps/wps_crop/process/crop/xml

Identifier	Title	Type	Identifier	Title
crop	crop image	ComplexData	Image	Image to crop
Abstract	Specification	Abstract		
crop a given image according to a set of coordinates	Process XML			

Previous Value: Pick the output from the previous link: b:Res

as the user connects or chains the services, the client tool automatically identifies the output from the first service and is recognised as the input for the second service.

Process URL: http://192.168.1.3:55000/wps/wps_crop/process/crop/xml

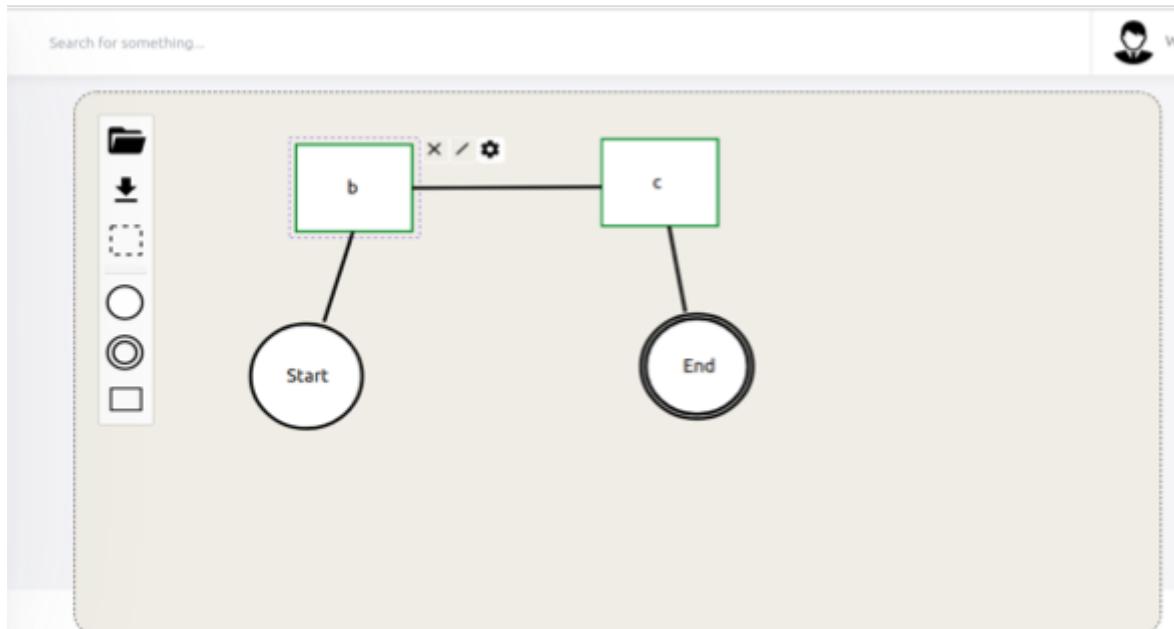
Identifier	Title	Type	Identifier	Title
crop	crop image	ComplexData	Image	Image to crop
Abstract	Specification	Abstract		
crop a given image according to a set of coordinates	Process XML			

Previous Value: b:Res

after configuring additional input parameters for the second service and the output parameters, the workflow is ready for validation.

the workflow is validated as a whole before the execution. The users are asked to validate the workflow by clicking the validate button. If the workflow is validated successfully the processing modules will turn green, see below:

if successful, an error message will appear, see below:



This allows the users to deploy an application to multiple preferred cloud providers. In this example, two cloud providers are shown as an example - AWS and Azure. The users will be presented with a Web Form to provide information regarding the clouds and the Workflow Engines.

Cloud Deployment Settings

Choose a Workflow Engine

Workflow Engine	Wefeos	⊕
-----------------	--------	---

Choose a Execution mode

Execution Mode		⊕
	Deploy and execute each application in separate cloud instance	
	Deploy and execute each application in the same cloud instance	
	Deploy and execute each application in the cloud instance (with more resources)	

Choose a Cloud Provider for each Wps

b	⊕
c	⊕

Choose a region to execute each Wps

As shown above, the user can choose a Workflow Engine. Currently only two types of engines are available - 1) Workflow Engine developed Internally developed 2) Camunda. The user can choose several options regarding the cloud deployment and execution. The options available are now: a) Deploy and execute each application in seprate cloud. In this case, first application is deployed in AWS and second application in Azure. b) Deploy and execute each application in same cloud c) Deploy and execute each application in cloud instance with more resources.

in the follow step, the user selects the preferred cloud provider for each service. See below:

Search for something... User icon

Cloud Deployment Settings

Choose a Workflow Engine

Workflow Engine	Wefeos	⊕
-----------------	--------	---

Choose a Execution mode

Execution Mode	Deploy and execute each application in separate cloud instance	⊕
----------------	--	---

Choose a Cloud Provider for each Wps

b	⊕
c	⊕
Amazon Web Services	
Microsoft Azure	

Choose a region to execute each Wps

after this configuration the user can execute the workflow. In this case,the client tool creates an instance in AWS, deploys the first application and executes. the below image shows the AWS account before the workflow execution(there is no available instance)

Search for something...

Execution Mode Deploy and execute each application in separate cloud instance

Choose a Cloud Provider for each Wps

- b Amazon Web Services
- c Microsoft Azure

Choose a region to execute each Wps

- b Frankfurt
- c Netherlands

Execute Workflow

The below image shows the same AWS account after the workflow execution

The screenshot shows the AWS EC2 Dashboard at the URL <https://eu-central-1.console.aws.amazon.com/ec2/v2/home?region=eu-central-1#instances:sort=instanceid>. The left sidebar menu is visible, showing options like EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES (with Instances selected), Launch Templates, Spot Requests, Reserved Instances, Dedicated Hosts, Capacity Reservations, IMAGES (with AMIs selected), and ELASTIC BLOCK STORE. The main content area displays a message: "You do not have any running instances in this region." Below it, there's a note: "First time using EC2? Check out the [Getting Started Guide](#)." A large blue "Launch Instance" button is prominently displayed. At the bottom, there are links for Feedback, English (US), and a footer with copyright information: "© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved." and links to Privacy Policy and Terms of Use.

after the execution of the first service in the workflow chain, the second service is deployed and executed in Azure in similar fashion. the below image shows the Azure account before the workflow execution (there is no available instance)

The screenshot shows the AWS EC2 Instances page. On the left sidebar, there are several navigation links: EC2 Dashboard, Events, Tags, Reports, Limits, Instances (selected), Launch Templates, Spot Requests, Reserved Instances, Dedicated Hosts, Capacity Reservations, AMIs, Bundle Tasks, and ELASTIC BLOCK STORE (Volumes, Snapshots). The main content area displays a table of instances. A single instance is listed: Name: b26d2607351849e183bcf61cd3332b36wpsbw, Instance ID: i-0a23311594eb01164, Instance Type: t2.micro, Availability Zone: eu-central-1a, Instance State: running, Status Checks: Initializing, Alarm Status: None, Public DNS (IPv4): ec2-54-93-177-62.eu-central-1.compute.amazonaws.com. Below the table, there's a detailed view for the selected instance. It shows the Instance ID (i-0a23311594eb01164), Public DNS (IPv4) (ec2-54-93-177-62.eu-central-1.compute.amazonaws.com), and the URL ec2-54-93-177-62.eu-central-1.compute.amazonaws.com. There are tabs for Description, Status Checks, Monitoring, and Tags. At the bottom of the page, there are links for Feedback, English (US), and a copyright notice: © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The below image shows the same AZURE account after the workflow execution

The screenshot shows the Azure Virtual Machines page. The top navigation bar includes a search bar, a user profile icon, and a 'DEFAULT DIRECTORY' dropdown. Below the navigation, the breadcrumb path is Home > Virtual machines. The main title is 'Virtual machines' with a 'Default Directory' link. There are buttons for Add, Reservations, Edit columns, Refresh, Assign tags, Start, Restart, Stop, Delete, and Services. A 'Subscriptions: Free Trial' message is displayed. Below the buttons, there are filters for 'Filter by name...', 'All resource groups', 'All types', 'All locations', 'All tags', and 'No grouping'. The table header includes columns for NAME, TYPE, STATUS, RESOURCE GROUP, LOCATION, MAINTENANCE STA..., and SUBSCRIPTION. The message '0 items' is shown above the table, followed by 'No results.' and a large grey placeholder icon. Below the icon, the text 'No virtual machines to display' is centered. At the bottom, there's a note: 'Create a virtual machine that runs Linux or Windows. Select an image from the marketplace or use your own customized image.' with links to 'Learn more about Windows virtual machines' and 'Learn more about Linux virtual machines'.

after the successful execution of the workflow, the first instance created in AWS has been terminated. see below

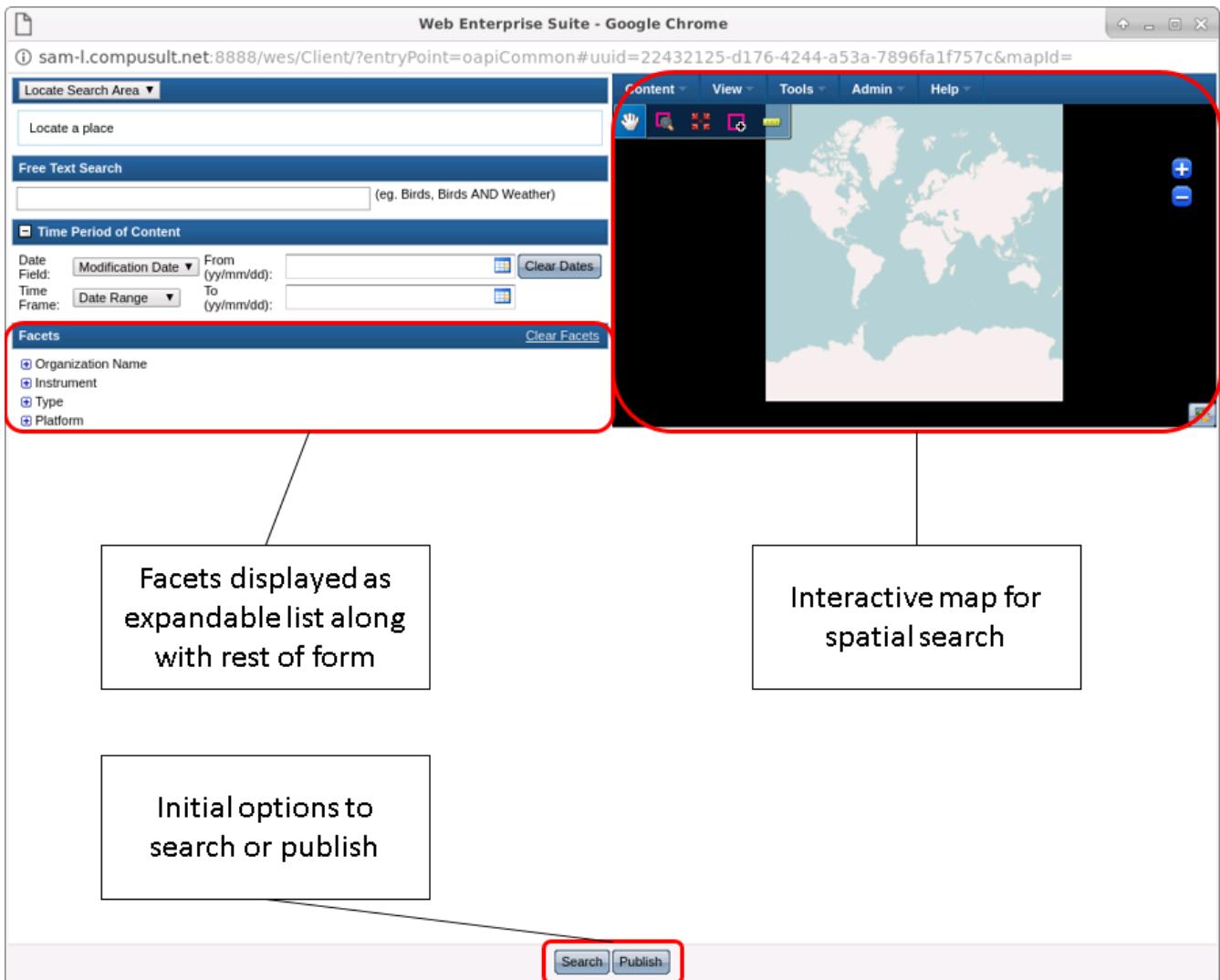
8.3.4. D123 - Client (Compusult)

For D123, Compusult delivered a Catalogue & Discovery Service Web Client. This client interacts with both [Catalog Server D121](#) and [Catalogue Server D117](#) as well as both D124 and [D118 WPS](#) instances and enables users to:

- Search both external catalogs;
- Register their applications in both catalogs;
- Update associated process/application metadata;
- Delete their applications from both catalogs;
- Deploy/Undeploy applications in a WPS-T (D118); and
- Build a service workflow or chain and display its results.

Searching External Catalogs

In order to search and present applicable search options (i.e. facets) from each external catalog, the client needs to retrieve various pieces of information. To do this, a request is made to the selected catalog for an OpenSearch Description Document (OSDD) (as mentioned in chapter 6.2.1). If the OSDD is available it is used to retrieve the query information and facets. If not, the client requests an SRU Explain Document (6.2.2) and uses that to obtain the facets and OpenAPI description (6.3.1) to obtain the query parameters. Once the information is retrieved, the facets are added to the search form and the OpenSearch query parameters are stored in memory and used to generate each appropriate search query. The search form allows users to specify free text, date ranges, area of interest and any advertised facets within their queries.



Upon selecting the search option, the client uses the specified criteria to generate the applicable search query which is then sent to the selected catalog.

Sample query ([Spacebel sample catalog](#)):

```
https://databio.spacebel.be/eo-catalog/resources?httpAccept=application/geo%2Bjson&query={free
text}&startRecord={page      size      x      (page      number-1)      +      1}&maximumRecords={page
size}&startDate={From                  date}&endDate={To                  date}&geometry={WKT
geometry}&organisationName={user      selected      value}&platform={user      selected
value}&instrument={user selected value}&type={user selected value}
```

The catalog returns the results as a GeoJSON FeatureCollection (as outlined in chapter 7.2.1), whose features array is then processed and presented as a list displaying simple information about each result in a user-friendly manner. The client takes the properties.title, properties.abstract and properties.updated tags from each feature for this purpose. Each result is accompanied by options to view the full metadata as well as management options (i.e. edit and delete) where applicable. The properties.identifier tag from each feature is stored in memory for the purposes of executing each of these service management queries at the user's discretion.

Web Enterprise Suite - Google Chrome

Locate Search Area ▾

Locate a place

Free Text Search (e.g. Birds, Birds AND Weather)

Time Period of Content

Date Field: Modification Date From (yy/mm/dd): Clear Dates

Time Frame: Date Range To (yy/mm/dd):

Facets

Organization Name

- 52 North (3)
- ATOS (1)
- CloudFerro (3)
- ESA/ESRIN (8)
- GLO-MERCATOR-TOULOUSE-FR (1)
- Sinergise (1)
- Spacebel (1)
- UHUL FMI (1)

Instrument

Type

Service management options for each result

Add to Workflow Delete Last Process Clear Workflow

Current Workflow:

BPMN Process Name: Publish Workflow

Sort By:

Calculation of NDVI using the SNAP toolbox for Sentinel-2 2019-06-05T00:00:00 -0230

 Calculation of NDVI using the SNAP toolbox for Sentinel-2 Data Scale:N/A



OGC WPS

Calculation of quality of an area of interest for Sentinel-2 2019-06-05T00:00:00 -0230

 Calculation of quality of an area of interest for Sentinel-2 Data Scale:N/A



OGC WPS

Showing 1 - 3 of 3

Page Size: 10 | Page 1 of 1

Search Publish



Service Management Interface

To allow users to register, edit and view their processes/applications, the client provided a special web-form.

Insert a New Application/Service - Google Chrome

sam-l.compusult.net:8888/wes/OAPISearchClient/pages/applicationForm.jsp?service=publish&catalog=http://databio.spacebel.be/eo-catalog/reso...

Identification Information

Abstract

Language

Publisher

Date Published

Version Info

Version Notes

Geometry

Contact Information

Links

Offerings

Submit

During registration, the web-form allows users to enter all applicable information to describe their process/application. Once complete, the client packages the information and sends the information to the catalog through a POST request to the /services path (as outlined in chapter 6.3.5).

To edit, the user simply executes a search to find the process/application to be updated. If the user does not have permission, they will only be able to view the metadata. If they have the applicable permissions, the client will provide an option to launch the web-form with all details auto-filled allowing them to edit the metadata. The updated metadata is sent via PUT request to the /services/{service-id} path.

When editing a service/application or viewing full metadata, the values of the fields are filled in based upon information found in the GeoJSON. Using the properties.identifier value, the client may send a GET request to the /services/{service-id} path to obtain the metadata about the service/application.

View Application/Service Metadata - Google Chrome

① sam-l.compusult.net:8888/wes/OAPISearchClient/pages/applicationForm.jsp?service=viewMetadata&id=MultiSensorNDVI&catalog=http://...

View Application/Service Metadata

[View GeoJSON](#)

Identification Information

Id (URI)

Identifier *

Title *

Date Updated *

Endpoint

Kind

NDVI is calculated after the two bands values Near Infrared and red. It is calculated by this formula : NDVI = (NIR-Red)/(NIR+Red)

Abstract

Language

Publisher

Date Published

Version Info

Version Notes

Geometry

First point will also be the last point of the polygon.

Long *
Lat *

Long *
Lat *

Long *
Lat *

...

Option to view metadata as raw GeoJSON

To remove a process/application from a catalog, users can select the associated delete option from the search results. Once confirmed, the applicable DELETE request is sent to the catalog via the /services/{service-id} path. The client will wait for confirmation from the catalog and display the appropriate message to the user.

Service Chaining

The client also features the ability to create a service chain or workflow based upon pre-existing services in a catalog.

A workflow may be created from two or more processes/applications registered within a catalog. When adding a new item, the client checks to ensure that at least one input of each new process matches the output of the last process in the workflow.

Once the user has completed their workflow, users have the option of deploying it to make it

available for discovery by other users. The client does this by converting the workflow to a BPMN document and deploying it to D124 (a WPS-T) which can then be registered within each external catalog.

The screenshot shows a web-based application interface titled "Web Enterprise Suite - Google Chrome". The left side features a search bar with fields for "Locate Search Area", "Free Text Search", and "Time Period of Content". Below these are "Facets" for "Organization Name", "Instrument", "Type", and "Platform". A large central area displays a world map. To the right, a red box highlights a workflow editor window. This window contains buttons for "Add to Workflow", "Delete Last Process", and "Clear Workflow". It shows a "Current Workflow" list with two items: "Calculation of quality of an area of interest for Sentinel-2" and "Calculation of NDVI using the SNAP toolbox for Sentinel-2". Below this is a "BPMN Process Name" input field and a "Publish Workflow" button. At the bottom of the main interface, there are sections for "Processing API at testbed.dev.52north.org" and "Rasdaman", each with a list of services and their details. The bottom right corner shows pagination controls for "Page Size: 10" and "Page 1 of 3".

Appropriate services can be selected to be added to workflow

Current Workflow:
Calculation of quality of an area of interest for Sentinel-2
Calculation of NDVI using the SNAP toolbox for Sentinel-2

BPMN Process Name: Publish Workflow

Processing API at testbed.dev.52north.org

- Calculation of NDVI using the SNAP toolbox for Sentinel-2
- Calculation of quality of an area of interest for Sentinel-2

Rasdaman

Rasdaman (raster data manager) is an open source array database system, which provides flexible, fast, scalable geo services for multi-dimensional spatio-temporal sensor, image, simulation, and statistics data of unlimited volume. Data is stored in a PostgreSQL database, thereby achieving full inf...

2019-06-05T00:00:00 -0230 OGC WPS

2019-06-04T00:00:00 -0230 OGC WPS

Showing 21 - 26 of 26

Page Size: 10 | Page 1 of 3

Interoperability tests

The technical integration experiments (TIEs) were done in stages. Initially, the client was tested using a [sample service provided by Spacebel](#). This service allowed testing many areas of the client as the service featured an OSDD, an OpenAPI description and an applicable GeoJSON search response. The service also featured the ability to retrieve individual applications or services which allowed for testing auto-populating our registration/edit web-form.

The ability to manage services was tested using the catalogs provided by GMU and Deimos. These catalogs featured the service management interface with the ability to add, edit and delete processes/applications or services allowing us to test and exercise all parts of the client.

Lessons learned

During the implementation of the Compusult client, Compusult discovered several areas that are potential pitfalls if not handled correctly. Compusult believes additional discussion in these areas is required in order to determine the correct path forward.

Authorization

Unfortunately, how the client should handle authorization while talking to each catalog was not clearly defined. This resulted in having to design special requests for each catalog. In the future, it would be beneficial for implementers if authorization was standardized and/or agreed upon during the earlier stages of development.

Service Management Interface

Designing a user-friendly application form was tricky given the fair amount of variance that can occur in the structure of any given GeoJSON feature. As outlined in the data model, as well as [OGC-14-055r2], an EOPAD GeoJSON feature object has a basic structure Compusult based their form around. However, any given feature could represent an EOPAD service/application, collection, or dataset. The data model also showed the ability to add wildcards to a large number of objects inside the structure of a feature. Developing a user-friendly form with such a large number of repeatable fields and the capability to publish different types of feature objects was complicated. Therefore, during the development of the form Compusult focused more on the essential parts outlined in the data model.

8.3.5. D118 - WPS Server (52°North)

The 52°North WPS Server implements the Web Processing Service 2.0 interface as well as the current draft version of the OGC API - Processes specification. As the current draft version of the OGC API - Processes specification does not specify a profile for transactional operations, a simple extension of the draft specification was developed based on application packages. For this, three new operations were added that support the creation, modification and deletion of processes (see [Table 44](#)). A complete specification in the form of an OpenAPI 3.0 document can be found in [Appendix E](#).

Table 44. Overview of the additionally transactional operations for the OGC API - Processes.

Method	URL	Description
POST	/processes	Deploy a new process by submitting an application package.
PUT	/processes/{:id}	Update the description of an existing process by submitting a application package.
DELETE	/processes/{:id}	Undeploy a process.

Processes published

Initially, the service did not offer any processes. However, two application packages were developed that could be deployed and chained. The first process (see [Example 3: Cloud coverage for an area of interest](#)) was able to calculate the cloud coverage of an area of interest of a Sentinel-2 scene, while the other calculated the Normalized Difference Vegetation Index (NDVI) of a Sentinel-2 scene (see [Example 2: NDVI Calculation](#)). Both processes were developed based on the [Sentinel Application Platform \(SNAP\)](#) [<http://step.esa.int/main/toolboxes/snap/>].

Catalog integration

The draft OGC API - Processes implementation is able to interface with one or more catalog services. The API registers the service at startup and deletes the service from the catalog when shutting down.

The initial description contains static operations as shown in table [Table 45](#).

Table 45. Initial operation offerings of the service.

Code	Method	URL	Description
LandingPage	GET	/	The landing page.
Service	GET	/api	The OpenAPI description.
Conformance	GET	/conformance	The list of conformance classes
Processes	GET	/processes	The process listing.
DeployProcess	POST	/processes	The endpoint to deploy new processes

Further, each process that is offered by the service is described in additional operations, see [Table 46](#).

Table 46. Operations for static processes.

Code	Method	URL	Description
DescribeProcess	GET	/processes/{:id}	The description of the process.
ExecuteProcess	POST	/processes/{:id}/jobs	The endpoint to execute a process.

When a new application package is deployed using the draft OGC API - Processes, the service checks if the application package is already registered in the catalog and registers the application if it is missing.

After this, the service description inside the catalog is updated to include the new process and a link to the application package with the relation type `hosts` is added. In addition to the operations that are also included for static, non-transactional processes, the description will contain additional operations for each transactional process (see [Table 47](#)).

Table 47. Operations for transactional processes.

Code	Method	URL	Description
UpdateProcess	PUT	/processes/{:id}	The endpoint to update the description of a process.
DeleteProcess	DELETE	/processes/{:id}	The endpoint to undeploy a process.

Similarly, when a process gets undeployed, the service description is updated to not include the

application anymore while the application remains in the catalog.

Identifier generation

The identifier of the service is generated from its URL by joining the reverse domain name with the dot-separated path elements of the URL. For example, <https://testbed.example.org/eopad/rest> will be registered as `org.example.testbed.eopad.rest`. Using this scheme, identifier collisions are unlikely. Special cases, such as <https://example.org/testbed/eopad/rest> would get the same identifier and would have to be minded by the domain owner.

Interoperability tests

The transactional OGC API - Processes successfully integrates with the catalog implementations of [Deimos \(D117\)](#) and [GMU \(D121\)](#). The service registers itself on startup and deregisters itself on shutdown. After the deployment of a new process, application packages are inserted into both catalogs if they are missing and the description of the service is amended inside the catalogs to reflect the newly deployed process. Correspondingly, the service description is updated to remove processes once they are undeployed.

8.3.6. D124 - WPS Server (Compusult)

For D124, Compusult delivered a Web Processing Service (WPS).

This WPS is registered in both catalogs D117 and D121.

Process Wrapper

To implement this item, Compusult used its transactional Web Processing Service (WPS-T).

Essentially, a process wrapper in the form of a Java servlet was created around NASA's Earth Observing System Data and Information System (EOSDIS) Gateway Interface (EGI) and then deployed to the Compusult's WPS-T. This made the process available via a WPS interface.

In order to access the EGI interface, a token is required. As part of our implementation, we generated a token which is pasted to the EGI from the WPS process wrapper each time a search is executed. Generated tokens are temporary and valid for 30 days.

The WPS process wrapper provides the ability to search the EGI using:

- Short Name of the collection to search (required)
- Time periods for date of creation and revision, in one of the following formats:
 - yyyy-MM-dd,yyyy-MM-dd
 - yyyy-MM-ddTHH:mm:ssZ,yyyy-MM-ddTHH:mm:ssZ
 - yyyy-MM-ddTHH:mm:ss,yyyy-MM-ddTHH:mm:ss
- Spatial data to specify points or areas of interest, in well-known text format (for example, "POLYGON 30 10, 40 40, 20 40, 10 20, 30 10")
- A generated token to execute a search (required)

An example search can be executed by creating a WPS execute request, supplying these parameters as strings with the identifiers “short_name”, “time”, “spatial” and “token”.

A full description of how to format a search for this process, including how to generate a token, exists at <https://ogc.compusult.com/wes/OAPISearchClient/pages/nasaegi.html>.

Results from executing this process are returned in a WPS response document featuring either a link to a zip folder if the search was successful or a short message explaining why the search failed.

An example execute is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Execute xmlns:wps="http://www.opengis.net/wps/2.0"
  xmlns:ows="http://www.opengis.net/ows/2.0" xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wps/2.0 ..../wpsExecute.xsd"
  service="WPS" version="2.0.0" response="document" mode="async">
  <ows:Identifier>EGI</ows:Identifier>
  <wps:Input id="short_name">
    <wps>Data>AE_Ocean</wps>Data>
  </wps:Input>
  <wps:Input id="time">
    <wps>Data>2010-06-06,2018-06-06</wps>Data>
  </wps:Input>
  <wps:Input id="spatial">
    <wps>Data>POINT (45 90)</wps>Data>
  </wps:Input>
  <wps:Input id="token">
    <wps>Data>your-token-here</wps>Data>
  </wps:Input>
  <wps:Output id="search_results" transmission="value"/>
</wps:Execute>
```

Interoperability tests

As mentioned in D123 above, the ability to manage services was tested using the catalogs provided by GMU and Deimos. Both of these catalogs provided the management facilities (i.e. add, edit and delete) to maintain client processes/applications. We used both of these catalogs to register, edit and remove the EGI WPS process.

In order to register a process/application with GMU's catalog, credentials were required. Once provided, they were added as part of each request in the basic HTTP authentication header.

8.3.7. DataBio Catalog Server (Spacebel)

Description

On behalf of the [H2020 DataBio](https://www.databio.eu/en/) [<https://www.databio.eu/en/>] project, Spacebel contributed a catalog server to the Testbed and OGC API Hackathon 2019 (See [\[OGC19-062\]](#)) implementing the EOPAD Catalogue service interface and the EOPAD data model for service/application metadata. It is an evolution of the ESA QC-MMS Catalog implementation to which support for service metadata has been added.

Interfaces implemented

The catalog server implements the OpenAPI description and draft OGC API Common interfaces described in the current Engineering Report. The server supports the OpenSearch [\[OGC13-026r8\]](#) discovery interface with JSON response as defined in [\[OGC17-047\]](#) including the faceted results. All interface documentation is accessible from the Landing Page.

<http://databio.spacebel.be/eo-catalog/> returns the Landing Page.

Landing Page response document

```
{  
  "links": [  
    {  
      "rel": "self",  
      "href": "http://databio.spacebel.be/eo-catalog",  
      "type": "application/json",  
      "title": "this document"  
    },  
    {  
      "rel": "service-desc",  
      "href": "http://databio.spacebel.be/eo-catalog/description",  
      "type": "application/openapi+json;version=3.0",  
      "title": "the OpenAPI definition"  
    },  
    {  
      "rel": "service-desc",  
      "href": "http://databio.spacebel.be/eo-catalog/description",  
      "type": "application/opensearchdescription+xml",  
      "title": "the OpenSearch Description Document"  
    },  
    {  
      "rel": "service-desc",  
      "href": "http://databio.spacebel.be/eo-catalog/description",  
      "type": "application/json;profile=\"http://explain.z3950.org/dtd/2.0/\"",  
      "title": "the Explain Document"  
    },  
    {  
      "rel": "service-doc",  
      "href": "http://databio.spacebel.be/eo-catalog/readme.html",  
      "type": "text/html",  
      "title": "the API Document"  
    },  
    {  
      "rel": "conformance",  
      "href": "http://databio.spacebel.be/eo-catalog/conformance",  
      "type": "application/json",  
      "title": "OGC conformance classes implemented by this API"  
    },  
    {  
      "rel": "data",  
      "href": "http://databio.spacebel.be/eo-catalog/collections",  
      "type": "application/json",  
      "title": "Metadata about the resource collections"  
    }  
  ]  
}
```

<http://databio.spacebel.be/eo-catalog/collections> returns the Collections metadata. Separate collections are used to group application/services metadata, EO collection metadata and EO product metadata. An overall collection **resources** groups all metadata.

Collections response document

```
{  
  "collections": [  
    {  
      "description": "All metadata records.",  
      "links": [  
        {  
          "rel": "items",  
          "href": "http://databio.spacebel.be/eo-catalog/resources",  
          "type": "application/geo+json",  
          "title": "Resources"  
        },  
        {  
          "rel": "search",  
          "href": "http://databio.spacebel.be/eo-  
catalog/description?httpAccept=application/opensearchdescription%2Bxml",  
          "type": "application/opensearchdescription+xml",  
          "title": "Search interface"  
        }  
      ],  
      "id": "resources",  
      "title": "Resources"  
    },  
    {  
      "description": "Metadata records representing EO services and  
applications.",  
      "links": [  
        {  
          "rel": "items",  
          "href": "http://databio.spacebel.be/eo-  
catalog/resources?type=service",  
          "type": "application/geo+json",  
          "title": "Services and applications"  
        },  
        {  
          "rel": "describedBy",  
          "href": "http://schemas.opengis.net/eopad-geojson/1.0/eopad-  
geojson-schema.json",  
          "type": "application/schema+json",  
          "title": "JSON schema for items belonging to this collection"  
        }  
      ],  
      "id": "services",  
      "title": "EO services and applications"  
    },  
    {
```

```

    "description": "Metadata records representing EO series (a.k.a. EO collections.)",
    "links": [
        {
            "rel": "items",
            "href": "http://databio.spacebel.be/eo-catalog/resources?type=collection",
            "type": "application/geo+json",
            "title": "Series"
        },
        {
            "rel": "describedBy",
            "href": "http://schemas.opengis.net/eoc-geojson/1.0/eoc-geojson-schema.json",
            "type": "application/schema+json",
            "title": "JSON schema for items belonging to this collection"
        }
    ],
    "id": "series",
    "title": "EO Series"
},
{
    "description": "Metadata records representing EO datasets (a.k.a. EO products.)",
    "links": [
        {
            "rel": "items",
            "href": "https://databio.spacebel.be/eo-catalog/resources/LANDSAT.ETM.GTC/datasets",
            "type": "application/geo+json",
            "title": "Datasets"
        },
        {
            "rel": "describedBy",
            "href": "http://schemas.opengis.net/eo-geojson/1.0/eo-geojson-schema.json",
            "type": "application/schema+json",
            "title": "JSON schema for items belonging to this collection"
        }
    ],
    "id": "datasets",
    "title": "EO Products"
}
],
"links": [
{
    "rel": "self",
    "href": "http://databio.spacebel.be/eo-catalog/collections",
    "type": "application/json",
    "title": "this document"
}
]

```

```
    ]  
}
```

The API definition is accessible via content negotiation in different formats. An **httpAccept query parameter** [http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/cs01/part3-sru2.0/searchRetrieve-v1.0-cs01-part3-sru2.0.html#_Toc324162475]¹ allows accessing the three response formats also from a Web browser:

- <http://databio.spacebel.be/eo-catalog/description?httpAccept=application/opensearchdescription%2Bxml>
- <http://databio.spacebel.be/eo-catalog/description?httpAccept=application/openapi%2Bjson;version=3.0>
- <http://databio.spacebel.be/eo-catalog/description?httpAccept=application/json;profile=%22http://explain.z3950.org/dtd/2.0/%22>

A pretty-printed version of the OpenAPI interface is available at <http://petstore.swagger.io/?url=https://databio.spacebel.be/eo-catalog/description?httpAccept=application/openapi%2Bjson;version=3.0>. Note that the OpenAPI definition includes security information declaring that the DELETE, PUT and POST methods are protected using **Basic Authentication** [<https://swagger.io/docs/specification/authentication/basic-authentication/>]. Protected methods are indicated with a lock symbol in the figure below.

Earth Observation Catalog

1.0.0 OAS3

<https://databio.spacebel.be/eo-catalog/description?httpAccept>

EO Catalog provides interoperable access, following ISO/OGC interface guidelines, to Earth Observation metadata. You can try all HTTP operations described in this OpenAPI 3.0 specification.

Spacebel s.a. - Website
Send email to Spacebel s.a.
Apache 2.0

Servers <https://databio.spacebel.be/eo-catalog> ▾ [Authorize](#)

LandingPage Provide Landing page of the catalog. ▾

GET / provide landing page (rel="self") in OAPI Common format

Conformance Provide conformance class information. ▾

GET /conformance provide conformance class information (rel="conformance") in OAPI Common format

APIDefinition Describe the catalogue interface. ▾

GET /resources/{seriesId}/description

GET /description Describe the catalogue interface (rel="service")

Collections Metadata about the resource collections. ▾

GET /collections Metadata about the resource collections

GET /collections/{collectionId} Metadata about a resource collection

Resources Operations for resources. ▾

POST /resources

GET /resources

Figure 48. DataBio Catalog Service OpenAPI interface

A readme page with several OpenSearch request examples is available at <http://databio.spacebel.be/eo-catalog/readme.html>. The readme page is depicted below.

Landing Page	http://databio.spacebel/eo-catalog/
Conformance	/conformance
Collections	/collections
(Collection) /collections/{collection-id}	<ul style="list-style-type: none"> • resources - /collections/resources • services - /collections/services • series - /collections/series
APIDefinition	<p>/description?httpAccept=application/opensearchdescription+xml (OpenSearch Description Document)</p> <p>/description?httpAccept=application/openapi+json;version=3.0 (OpenAPI Definition)</p> <p>[Open with Redoc Open with Swagger.io]</p> <p>/description?httpAccept=application/json;profile="http://explain.z3950.org/dtd/2.0/" (Explain Document)</p>
(Services) /resources	<ul style="list-style-type: none"> • DataBio (Filter by dc:type, dc:subject) - /resources?type=service&subject=databio • DataBio (Filter by os:startIndex, os:count) - /resources?type=service&subject=databio&startRecord=2&maximumRecords=1 • Testbed-15 (Filter by dc:subject) - /resources?type=service&subject=testbed-15 • Testbed-15 (Filter by dc:type, eo:organisationName) - /resources?type=service&organisationName="52%20North" • Testbed-15 (Filter by eo:offering) - /resources?type=service&offering=docker
(Service) /resources/{service-id}	<ul style="list-style-type: none"> • DataBio Component C07.01 - /resources/59c264f8e4b00685883826c7 • DataBio Component C05.01 - /resources/59c3b4fae4b00685883826d7 • DataBio Component C14.01 - /resources/59ce3e48e4b006858838270d • Testbed-15 NdviCalculation - /resources/org.n52.project.tb15.eopad.NdviCalculation • Testbed-15 SentinelQualityAreaOfInterest - /resources/org.n52.project.tb15.eopad.SentinelQualityAreaOfInterest • Testbed-15 MultiSensorNDVI - /resources/MultiSensorNDVI • Testbed-15 WPS - /resources/testbed.dev.52north.org.javaps-eopad.rest
(Series) /resources	<ul style="list-style-type: none"> • Testbed-14 (Filter by dc:subject) - /resources?type=collection&subject=testbed-14 • DataBio (Filter by os:searchTerms) - /resources?type=collection&query=Copernicus
(Series) /resources/{series-id}	<ul style="list-style-type: none"> • Sentinel-2 - /resources/EOP:ESA.Sentinel-2 • Landsat-5 - /resources/LANDSAT.TM.GTC • Landsat-7 - /resources/LANDSAT.ETM.GTC • Synergiqe Sentinel2 - /resources/EOP:SENTINEL-HUB.Sentinel2 • Testbed-14 IPT:Sentinel2 - /resources/EOP:IPT.Sentinel2 • Testbed-14 IPT:Landsat8 - /resources/EOP:IPT.Landsat8 • Testbed-14 DE2_MS4_L1B - /resources/DE2_MS4_L1B

Figure 49. Readme page

Metadata published

The catalog hosts GeoJSON metadata for EO applications and services, EO collections (OGC 17-084) and EO product (OGC 17-003). Metadata hosted includes application metadata from DataBio Hub (<https://www.databiohub.eu/registry/>) and collection metadata originating from DataBio and OGC Testbed-14 and Testbed-15. Application and service metadata from participants produced as part of this testbed is available as well. The metadata can be found with the requests below:

- <https://databio.spacebel.be/eo-catalog/resources?organisationName=52%20North>
- <https://databio.spacebel.be/eo-catalog/resources?organisationName=Compusult>

Applications and services which are available as Docker containers can be retrieved as follows:

- <https://databio.spacebel.be/eo-catalog/resources/?type=service&offering=docker>

The maximum number of facet counts returned for a facet (e.g. eo:organisationName) can be limited by a client using the sru:facetLimit OpenSearch parameter as follows:

- <http://databio.spacebel.be/eo-catalog/resources?type=service&facetLimit=eo:organisationName:2>
- <http://databio.spacebel.be/eo-catalog/resources?type=collection&facetLimit=eo:platform:3>

Metadata files can be added to the catalog using a command as below:

```
curl -H "Content-type: application/geo+json" --data-binary @metadata.json  
http://user:password@data.bio.spacebel.be/eo-catalog/resources
```

Interoperability tests

The catalog was mainly used for early interoperability tests by client developers, such as the [Compusult Client D123](#) as described in the sections above.

As part of the OGC API Hackaton, additional interoperability tests were done with the OGC API Features client implementation provided by Sigma Bravo based on the OpenSphere client software (<http://frozen-lime.surge.sh/>).

Lessons learned

Spacebel believes that there is further work required to fully harmonise the OGC API Common/Features specifications and current OGC OpenSearch standards (including OGC 10-032r8 and OGC 17-047) and GeoJSON features based on OGC14-055r2 to facilitate the coexistence and interoperability of both types of OGC interfaces and GeoJSON features. Aligning OGC14-055r2 with the OGC API Features Core standard (e.g. Link model and location of links inside a FeatureCollection and/or Feature object) would be a good starting point according to Spacebel. The update of OGC14-055r2 could be used as an opportunity to align some of the property names in this standard with existing DCAT property names and deprecate some of the original property names.

The flexibility and freedom that current catalogue server implementers have to decide on HTTP query parameter names when implementing OGC OpenSearch interfaces while remaining interoperable, e.g. one may have chosen to comply with OASIS searchRetrieve standard parameter names, is lost when migrating to OGC API Common compatible interfaces as the query parameter names become fixed in the OpenAPI description.

Future work

The Catalog server could decide based on the identity and rights of the user to allow or not addition (POST), removal (DELETE) or updates (PUT) of metadata records in the server. This could be done by mandating the server to include the [Allow header field](#) [<https://tools.ietf.org/html/rfc7231#section-7.4.1>] in the HTTP header of the response to advertise the supported HTTP methods on a specific resource. Catalog clients can then present a proper user interface to the user taking into account the available methods on each resource as advertised by the server. Implementing the [HEAD](#) method would allow a client to request the allowed methods for an individual resource also. This approach is also used by [W3C LDP 1.0](#) RESTful interfaces.

Chapter 9. Arctic Discovery Catalogue



9.1. Overview

The Arctic Discovery Catalogue component (D107) was built to offer an evergreen catalogue of relevant Arctic circumpolar Web services from OGC standards enabled services and Esri REST Web services that have some relevance to circumpolar science. The component was built to evaluate the confidence level of the web services, and to make the results available via an OGC Catalogue Service implementation. [Figure 48](#) illustrates the relationship between this component and related Testbed-15 deliverables.



Figure 50. D107 Overview

9.2. Architecture

The architecture of the Arctic Discovery Catalogue is as follows:



1. The Discovery and Machine Learning (ML) Engine trains the ML model using keywords, geographic areas, and positive/negative training datasets. This process is described in the Testbed-15 D002 Machine Learning ER.
2. The Discovery and ML Engine runs in the background, harvests and processes each configured endpoint and populates the Compusult Web Enterprise Suite (WES) CSW ebRIM data model with items that match the model
3. Users use the CSW Client to query the Arctic relevant content of the CSW

A sequence diagram of the interactions between these components is shown in [Figure 49](#).

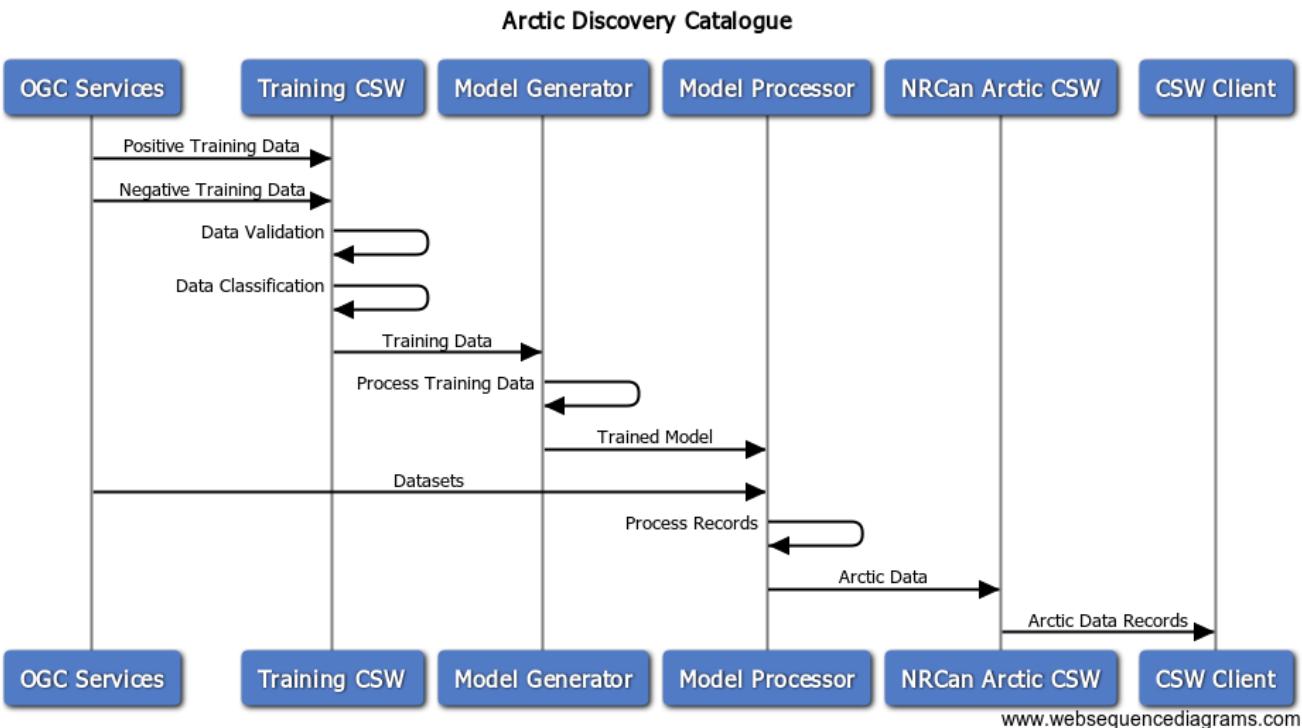


Figure 51. Sequence Diagram

1. Both positive and negative training data is harvested from various OGC enabled Data Services into the Training CSW
2. The Training Data is validated
3. The Training Data is classified
4. The Model Generator queries the training data from the Training CSW
5. The Model Generator processes the training data and builds the ML Model
6. The Model Processor retrieves records from various online OGC Services and applies the trained model to filter the results
7. The filtered results are populated into an NRCan Arctic CSW
8. Clients can access the NRCan Arctic CSW to retrieve the Arctic only records

9.3. Training CSW

The initial task of the Arctic Discovery Catalogue was to populate the training dataset which would be used to train the ML model. Compusult stood up a blank WES CSW and this Training CSW was populated with both positive and negative training data harvested from a variety of CSW data sources.

Table 48. CSW Training Datasets

Dataset	Description
<i>Catalogues</i>	<p>Positive:</p> <p>Arctic SDI Portal [https://geoportal.arctic-sdi.org], CGDI Geospatial Web Services [https://www.nrcan.gc.ca/earth-sciences/geomatics/canadas-spatial-data-infrastructure/19359], Atlas of the Cryosphere: Northern Hemisphere [https://nsidc.org/cgi-bin/atlas_north?service=WMS&request=GetCapabilities&version=1.1.1], Harvested Conservation of Arctic Flora and Fauna (CAFF) abds [http://geo.abds.is/geoserver/base/wms?Request=GetCapabilities&version=1.3.0], Harvested ArcticSDI_US ArcGIS service [https://services.nationalmap.gov/arcgis/rest/services/ArcticSDI_US/MapServer], Harvested Canadian Cryospheric Information Network [https://ccin.ca], Polar Data Catalogue [https://polardata.ca]</p> <p>Negative:</p> <p>Geoscience Australia Product Catalogue [https://ecat.ga.gov.au/geonetwork/srv/eng/csw?request=GetCapabilities&service=CSW], Afghanistan Disaster Risk WebGIS [http://disasterrisk.af/catalogue/csw?service=CSW&version=2.0.28&request=GetCapabilities], Tuna atlas [http://tunaatlas.d4science.org/geonetwork/srv/eng/csw?service=CSW&request=GetCapabilities]</p>

Once this data was harvested the Training CSW contained over 54,000 records.

```
<?xml version="1.0" encoding="UTF-8"?>
<csw:GetRecordsResponse xmlns:csw="http://www.opengis.net/cat/csw/2.0.2" xmlns:dc=
"http://purl.org/dc/elements/1.1/" xmlns:dct="http://purl.org/dc/terms/" xmlns:gml=
"http://www.opengis.net/gml" xmlns:ows="http://www.opengis.net/ows" xmlns:rim=
"urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0" xmlns:wrs=
"http://www.opengis.net/cat/wrs/1.0" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xml=
"http://www.w3.org/XML/1998/namespace" xsi:schemaLocation=
"http://www.opengis.net/cat/csw/2.0.2 http://schemas.opengis.net/csw/2.0.2/CSW-
discovery.xsd">
    <csw:SearchStatus timestamp="2019-06-07T18:22:32"/>
    <csw:SearchResults elementSet="full" nextRecord="1" numberOfRecordsMatched="64575"
numberOfRecordsReturned="0"/>
</csw:GetRecordsResponse>
```

9.4. Evergreen Catalogue

To facilitate the Evergreen capability of the catalogue, Compusult has implemented the Harvest interface of the CSW specification. The Harvest operation defines an interface for indirectly creating, modifying and deleting catalogue records by invoking a CSW client harvesting run from the server to a specified target. The harvesting operation can be run in either synchronous or asynchronous mode and can be configured to execute just once or periodically. This operation requires user authentication to be invoked.

The Harvest operation can be invoked via a POST request or a GET request as follows:

Synchronous one-run Harvest example

POST Request

```
Url: http://localhost:8080/geonetwork/srv/eng/csw-publication  
Content-type: application/xml
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<csw:Harvest xmlns:csw="http://www.opengis.net/cat/csw/2.0.2" xmlns:gmd=  
"http://www.isotc211.org/2005/gmd" service="CSW" version="2.0.2">  
    <csw:Source>http://[ URL to the target CSW server ]?request=GetCapabilities  
&service=CSW&version=2.0.2</csw:Source>  
    <csw:ResourceType>http://www.isotc211.org/schemas/2005/gmd/</csw:ResourceType>  
</csw:Harvest>
```

GET Request

```
Url: http://localhost:8080/geonetwork/srv/eng/csw-  
publication?request=Harvest&service=CSW&version=2.0.2&Source=http://[ URL to the  
target CSW server ]&ResourceType=http://www.isotc211.org/schemas/2005/gmd/
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>  
<csw:HarvestResponse xmlns:csw="http://www.opengis.net/cat/csw/2.0.2">  
    <csw:TransactionResponse>  
        <csw:TransactionSummary>  
            <csw:totalInserted>98</csw:totalInserted>  
            <csw:totalUpdated>14</csw:totalUpdated>  
            <csw:totalDeleted>0</csw:totalDeleted>  
        </csw:TransactionSummary>  
    </csw:TransactionResponse>  
</csw:HarvestResponse>
```

Asynchronous one-run Harvest example

POST Request

```
Url: http://localhost:8080/geonetwork/srv/eng/csw-publication  
Content-type: application/xml
```

```

<?xml version="1.0" encoding="UTF-8"?>
<csw:Harvest xmlns:csw="http://www.opengis.net/cat/csw/2.0.2" xmlns:gmd=
"http://www.isotc211.org/2005/gmd" service="CSW" version="2.0.2">
    <csw:Source>http://[ URL to the target CSW server ]?request=GetCapabilities
&service=CSW&version=2.0.2</csw:Source>
    <csw:ResourceType>http://www.isotc211.org/schemas/2005/gmd/</csw:ResourceType>
    <csw:ResponseHandler>[ URI or email address of response handler
]</csw:ResponseHandler>
</csw:Harvest>

```

GET Request

Url: http://localhost:8080/geonetwork/srv/eng/csw-publication?request=Harvest&service=CSW&version=2.0.2&Source=http://[URL to the target CSW server]&ResourceType=http://www.isotc211.org/schemas/2005/gmd/&ResponseHandler=[URI or email address of response handler]

Response

```

<?xml version="1.0" encoding="UTF-8"?>
<csw:HarvestResponse xmlns:csw="http://www.opengis.net/cat/csw/2.0.2">
    <csw:Acknowledgement timeStamp="2011-12-05T15:13:59">
        <csw:EchoedRequest>
            <csw:Harvest xmlns:csw="http://www.opengis.net/cat/csw/2.0.2" xmlns:gmd=
"http://www.isotc211.org/2005/gmd" service="CSW" version="2.0.2">
                <csw:Source>http://[ URL to the target CSW server
]?request=GetCapabilities&service=CSW&version=2.0.2</csw:Source>
                <csw:ResourceType>
http://www.isotc211.org/schemas/2005/gmd/</csw:ResourceType>
                <csw:ResponseHandler>[ URI or email address of response handler
]</csw:ResponseHandler>
            </csw:Harvest>
        </csw:EchoedRequest>
        <csw:RequestId>dfc5e552-93a2-4150-a332-c5c1de5857c1</csw:RequestId>
    </csw:Acknowledgement>
</csw:HarvestResponse>

```

Source: [geonetwork-opensource.org](https://geonetwork-opensource.org/manuals/2.10.4/eng/developer/xml_services/csw_services.html#harvest) [https://geonetwork-opensource.org/manuals/2.10.4/eng/developer/xml_services/csw_services.html#harvest]

9.5. WES Sync

As a result of the success of this Testbed, Compusult enhanced it's Web Enterprise Suite (WES) product to add a Sync feature which will synchronize two WES CSW Catalogues using filters based on two approaches:

1. Customizable Boolean filter - Using this approach records are filtered by specifying criteria in a

"SQL WHERE Clause" format so that records are filtered based on the specified criteria during a sync operation.

2. Machine Learning Model filter - Using this approach records are filtered using a machine learning model.

Appendix A: Abstract Test Suite

Future discussion of this document within the OGC Standards Program should consider development of an Abstract Test Suite.

Appendix B: Interpreting JSON as JSON-LD

The [JSON-LD] encoding proposed in this annex supports a seamless transition from [GeoJSON] based implementations to JSON-LD based implementations with a minimum number of edits. The GeoJSON encoding proposed can be interpreted as JSON-LD by JSON-LD-aware implementations through the use of the normative @context document provided in this annex.

B.1. Introduction

B.1.1. JSON-LD

The serialized GeoJSON form of an EOPAD document shall be consistent with what would be produced by the standard JSON-LD 1.0 Processing Algorithms and API [JSON-LD-API] Compaction Algorithm using, at least, the normative JSON-LD @context definition provided in Annex B.2. Implementations may augment the provided @context with additional @context definitions but shall not override or change the normative context. Implementations may also use additional properties and values not defined in the JSON-LD @context with the understanding that any such properties will likely be unsupported and ignored by consuming implementations that use the standard JSON-LD algorithms. See the Extensibility section for more information on handling extensions within EOPAD documents.

JSON-LD uses the special @context property to define the processing context. The value of the @context property is defined by the JSON-LD specification. Implementations producing EOPAD metadata documents should include a @context property with a value that includes a reference to the normative JSON-LD @context definition using the URL <https://www.opengis.net/spec/eopad-geojson/1.0>. Implementations may use the alternative URL <http://www.opengis.net/spec/eopad-geojson/1.0> instead.

When a JSON-LD enabled EOPAD metadata implementation encounters a GeoJSON document identified using the `application/geo+json` MIME media type, and that document does not contain a @context property whose value includes a reference to the normative JSON-LD @context definition, the implementation shall assume that the normative @context definition still applies.

Alternatively, implementations can refer to the normative JSON-LD context document via an HTTP Link header as described in section 6.8 of the JSON-LD specification [JSON-LD]. This allows the GeoJSON documents to be unambiguously machine readable as JSON-LD.

Implementations can also natively implement JSON-LD representations instead of the recommended GeoJSON representations. The corresponding resource specification diagram then looks as depicted below.

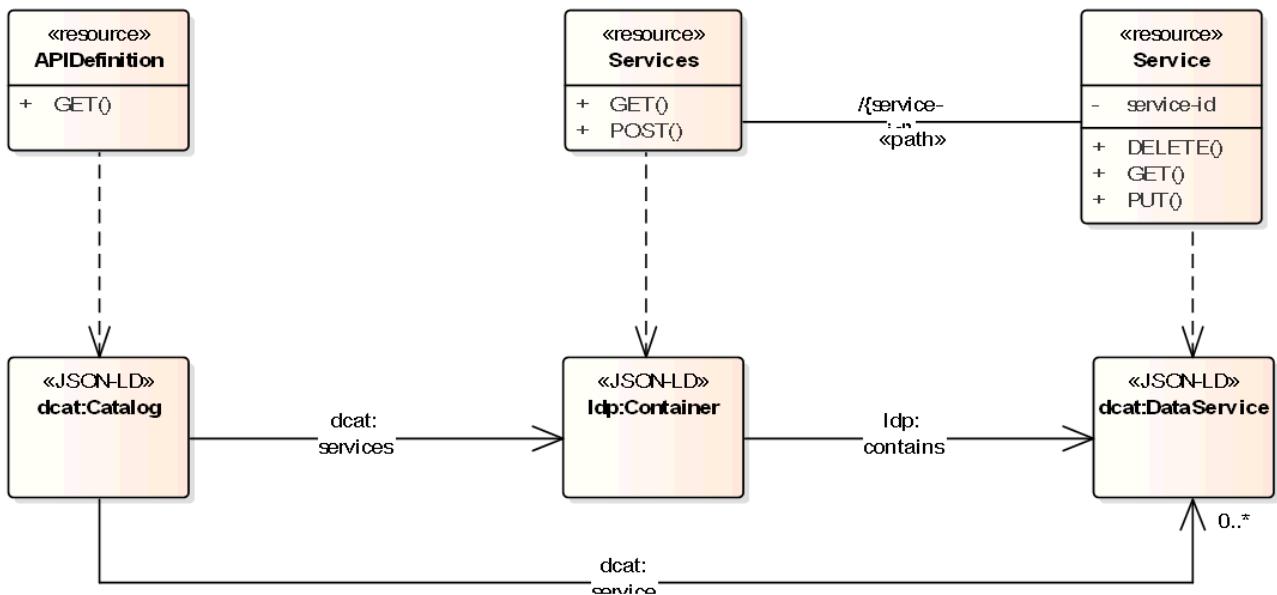


Figure 52. Resource Specification Diagram (JSON-LD Representations)

JSON-LD aware tools (e.g. [JSON-LD Playground](https://json-ld.org/playground/) [<https://json-ld.org/playground/>]) can interpret an EOPAD GeoJSON metadata record (Feature) or catalogue response (FeatureCollection) as a JSON-LD **dcat:DataService** and **ldp:Container** respectively through the normative context. This is depicted below.

The screenshot shows the JSON-LD Playground interface with the following details:

- Top Bar:** Examples: Person, Event, Place, Product, Recipe, Library, Activity. Buttons for Permalink, Gist, and Shortcuts.
- Input Area:** A text input field containing a GeoJSON feature object. The JSON structure includes properties like geometry, id, type, properties, kind, theme, and links.
- Output Area:** A large text area showing the resulting JSON-LD representation. It includes triples such as `http://www.w3.org/ns/ldp#contains`, `http://www.w3.org/ns/dcat#contactPoint`, and `http://purl.org/dc/terms/hasPart`.
- Toolbars:** Below the input and output areas are various toolbar buttons for expanding, compacting, flattening, framing, and visualizing the JSON-LD structure.

Figure 53. JSON-LD Playground

B.1.2. Semantic Web and JSON-LD

The W3C Semantic Web Activity is defining a collection of technologies that enables a "Web of data" where information is easily shared and reused across applications. Some key pieces of this technology stack are the RDF (Resource Description Framework), RDFS (RDF Schema) data models and JSON-LD. The JSON-LD encoding proposed in the current document is a concrete RDF syntax and can be interpreted as the proposed encoding in RDF for EO applications and related processing services metadata. Each JSON(-LD) property proposed corresponds to an RDF predicate which is represented with a URI. The relationship between JSON-LD and RDF is explained [here](https://www.w3.org/TR/json-ld/#relationship-to-rdf) [https://www.w3.org/TR/json-ld/#relationship-to-rdf].

All proposed properties are defined (via a normative @context document) with an URI from existing, well-known, open vocabularies where available, e.g. Dublin Core (dct:), DCAT (dcat:), OWL (owl:), ADMS (adms:), PROV (prov:), vcard (vcard:) etc.

The RDF (JSON-LD) encoding proposed in the current document for EOPAD metadata can therefore be stored in an RDF triple-store and made accessible through a (Geo)SPARQL [\[OGC11-052r4\]](#) endpoint as well.

B.1.3. JSON-LD and Structured Data

Major search engines are able to extract JSON-LD markup (structured data) embedded in <script> tags in the Web page head or body. For instance, [Google](#) [https://ai.googleblog.com/2017/01/facilitating-discovery-of-public.html] supports both schema.org and W3C DCAT properties. In the current document, we make heavy use of the W3C DCAT vocabulary.

This eventually should allow the proposed EOPAD metadata files published on a Web server, or important parts of it, to be picked up by mass market search engine crawlers. A similar JSON-LD @context mapping to schema.org, possibly using the mapping proposed in section 11.1 of [\[DCAT-2018\]](#) or [Annex B: Alignment with Schema.org"](#) [https://w3c.github.io/dxwg/dcat/#dcat-sdo] of [\[DCAT-2019\]](#), is possible and presented in the section [Relation with Schema.org](#) below. As Testbed-15 has selected [\[DCAT\]](#) as its primary vocabulary, further refinement of the alternative mapping with schema.org is left as future work.

B.2. JSON-LD @context definition

B.2.1. JSON-LD 1.1

The current annex provides the normative JSON-LD @context which is implied by the GeoJSON encoding as explained above. It is available at <http://schemas.opengis.net/eopad-geojson/1.0/eopad-geojson.jsonld>. JSON-LD 1.1 aware clients can apply a JSON-LD 1.1 @context to interpret the GeoJSON encoding as JSON-LD.

```
{  
  "@context": "http://schemas.opengis.net/eopad-geojson/1.0/eopad-geojson.jsonld"  
}
```

The context relies on [JSON-LD 1.1](#) specific features, in particular „scoped contexts“ and „nested properties“. The expressiveness of JSON-LD 1.0 @context is too limited to isolate the required context information to support compacting and expansion in a single external context document. It would require inserting inline contexts in the GeoJSON encoding.

eopad-geojson.jsonld

```
{  
  "@context": {  
    "@version": 1.1,  
    "dct": "http://purl.org/dc/terms/",  
    "atom": "http://www.w3.org/2005/Atom/",  
    "iana": "http://www.iana.org/assignments/relation/",  
    "os": "http://a9.com/-/spec/opensearch/1.1/",  
    "eop": "http://www.opengis.net/ont/eo-geojson/1.0/",  
    "owc": "http://www.opengis.net/ont/owc/1.0/",  
    "gj": "https://purl.org/geojson/vocab#",  
    "gsp": "http://www.opengis.net/ont/geosparql#",  
    "vcard": "http://www.w3.org/2006/vcard/ns#",  
    "skos": "http://www.w3.org/2004/02/skos/core#",  
    "dcat": "http://www.w3.org/ns/dcat#",  
    "xsd": "http://www.w3.org/2001/XMLSchema#",  
    "prov": "http://www.w3.org/ns/prov#",  
    "locn": "http://www.w3.org/ns/locn#",  
    "foaf": "http://xmlns.com/foaf/0.1/",  
    "schema": "http://schema.org/",  
    "rdfs": "http://www.w3.org/2000/01/rdf-schema#",  
    "adms": "http://www.w3.org/ns/adms#",  
    "owl": "http://www.w3.org/2002/07/owl#",  
    "wps": "http://www.opengis.net/wps/2.0/",  
    "ldp": "http://www.w3.org/ns/ldp#",  
  
    "title": "dct:title",  
    "identifier": "dct:identifier",  
    "issued": "dct:issued",  
    "rights": "dct:rights",  
    "license": {  
      "@id": "dct:license",  
      "@context": {  
        "label": "rdfs:label"  
      }  
    },  
    "LicenseDocument": "dct:LicenseDocument",  
    "accessRights": {  
      "@id": "dct:accessRights",  
      "@context": {  
        "label": "rdfs:label"  
      }  
    },  
    "RightsStatement": "dct:RightsStatement",  
  }  
}
```

```

"provenance": {
    "@id": "dct:provenance",
    "@context": {
        "label": "rdfs:label"
    }
},
"ProvenanceStatement": "dct:ProvenanceStatement",

"wasUsedBy": {
    "@id": "prov:wasUsedBy",
    "@context": {
        "generated": "prov:generated",
        "description": "dct:description",
        "qualifiedAssociation": "prov:qualifiedAssociation",
        "degree": {
            "@id": "dct:type",
            "@type": "@id"
        }
    }
},
"hadPlan": {
    "@id": "prov:hadPlan",
    "@context": {
        "wasDerivedFrom": "prov:wasDerivedFrom"
    }
},
"Activity": "prov:Activity",
"Association": "prov:Association",
"Entity": "prov:Entity",
"Plan": "prov:Plan",
"Standard": "dct:Standard",

"lang": {
    "@id": "dct:language",
    "@type": "@id",
    "@context": {
        "@base": "http://id.loc.gov/vocabulary/iso639-1/"
    }
},
"hreflang": {
    "@id": "dct:language",
    "@type": "@id",
    "@context": {
        "@base": "http://id.loc.gov/vocabulary/iso639-1/"
    }
},
"kind": {
    "@id": "dct:type",
    "@type": "@id"
}

```

```
},  
  
"doi": {  
    "@id": "adms:identifier",  
    "@type": "@id",  
    "@context": {  
        "@base": "https://doi.org/"  
    }  
},  
  
"isPrimaryTopicOf": "foaf:isPrimaryTopicOf",  
  
"updated": "dct:modified",  
"published": "dct:issued",  
  
"versionInfo": "owl:versionInfo",  
"versionNotes": "adms:versionNotes",  
"bibliographicCitation": "dct:bibliographicCitation",  
"creator": "dct:creator",  
"abstract": "dct:description",  
"subtitle": "dct:description",  
"subject": "dct:subject",  
"hasPart": {  
    "@id": "dct:hasPart",  
    "@type": "@id"  
},  
"date": "dct:date",  
"Location": "dct:Location",  
"spatial": {  
    "@id": "dct:spatial",  
    "@context": {  
        "type": "@type",  
        "value": "@value",  
        "id": "@id",  
        "geometry": "locn:geometry"  
    }  
},  
"keyword": "dcat:keyword",  
"categories": "dcat:theme",  
"distribution": {  
    "@id": "dcat:distribution",  
    "@context": {  
        "label": "rdfs:label"  
    }  
},  
"Distribution": "dcat:Distribution",  
"endpointDescription": {  
    "@id": "dcat:endpointDescription",  
    "@type": "@id"  
},  
"links": {
```

```

"@id": "owc:links",
"@context": {
    "@vocab": "http://www.iana.org/assignments/relation/",
    "type": "atom:type",
    "data": "iana:enclosure",
    "profiles": "iana:profile",
    "previews": "iana:icon",
    "alternates": "iana:alternate",
    "qualityReport": "iana:describedby"
},
},

"Links": "owc:Links",
"offerings": {
    "@id": "dcat:endpointDescription",
    "@context": {
        "@vocab": "http://www.opengis.net/ont/owc/1.0/",
        "code": {
            "@id": "owc:code",
            "@type": "@id"
        }
    }
},
"operations": {
    "@id": "owc:operations",
    "@context": {
        "href": "owc:href",
        "code": "owc:code",
        "type": "owc:type"
    }
},
"contents": {
    "@id": "owc:contents",
    "@context": {
        "type": "owc:type"
    }
},
"processDescription": {
    "@id": "wps:processDescription",
    "@context": {
        "@vocab": "http://www.opengis.net/wps/2.0/"
    }
},
"Link": "atom:link",
"href": "@id",
"beginningDateTime": "schema:startDate",
"endingDateTime": "schema:endDate",
"length": "atom:length",
"Category": "skos:Concept",

```

```

"label": "skos:prefLabel",
"term": "@id",
"schema": "skos:inScheme",
"hasGeometry": "gsp:hasGeometry",
"asWKT": "gsp:asWKT",
"Feature": "dcat:DataService",
"FeatureCollection": "ldp:Container",
"GeometryCollection": "gj:GeometryCollection",
"LineString": "gj:LineString",
"MultiLineString": "gj:MultiLineString",
"MultiPoint": "gj:MultiPoint",
"MultiPolygon": "gj:MultiPolygon",
"Point": "gj:Point",
"Polygon": "gj:Polygon",
"bbox": {
    "@container": "@list",
    "@id": "gj:bbox"
},
"coordinates": "gj:coordinates",
"features": {
    "@container": "@set",
    "@id": "ldp:contains"
},
"geometry": "gj:geometry",
"properties": "@nest",
"acquisitionInformation": "@nest",
"acquisitionParameters": "dct:temporal",
"Platform": "eop:Platform",
"platform": {
    "@id": "eop:platform",
    "@context": {
        "@vocab": "http://www.opengis.net/ont/eo-geojson/1.0/"
    }
},
"Instrument": "eop:Instrument",
"instrument": {
    "@id": "eop:instrument",
    "@context": {
        "@vocab": "http://www.opengis.net/ont/eo-geojson/1.0/"
    }
},
"id": "@id",
"type": "@type",
"authors": {
    "@id": "dct:creator",
    "@context": {
        "@vocab": "http://xmlns.com/foaf/0.1/",
        "Kind": "foaf:Agent",
        "Individual": "foaf:Person",
        "email": "foaf:mbox",
        "name": "foaf:name",
}
}

```

```

        "uri": {
            "@id": "foaf:page",
            "@type": "@id"
        }
    },
    "Document": "foaf:Document",
    "page": "foaf:page",
    "uri": "@id",
    "qualifiedAttribution": {
        "@id": "prov:qualifiedAttribution",
        "@context": {
            "@vocab": "http://www.w3.org/2006/vcard/ns#",
            "Attribution": "prov:Attribution",
            "agent": "prov:agent",
            "role": {
                "@id": "dct:type",
                "@type": "@id",
                "@context": {
                    "@base": "http://inspire.ec.europa.eu/metadata-
codeList/ResponsiblePartyRole/"
                }
            },
            "Agent": "vcard:Kind",
            "Person": "vcard:Individual",
            "email": "vcard:hasEmail",
            "name": "vcard:hasName",
            "phone": "vcard:hasTelephone",
            "uri": {
                "@id": "vcard:hasURL",
                "@type": "@id"
            }
        }
    },
    "contactPoint": {
        "@id": "dcat:contactPoint",
        "@context": {
            "@vocab": "http://www.w3.org/2006/vcard/ns#",
            "Agent": "vcard:Kind",
            "Person": "vcard:Individual",
            "email": "vcard:hasEmail",
            "name": "vcard:hasName",
            "phone": "vcard:hasTelephone",
            "uri": {
                "@id": "vcard:hasURL",
                "@type": "@id"
            }
        }
    }
}

```

B.2.2. Relation with DCAT and GeoDCAT-AP

The JSON-LD representation obtained by expanding the GeoJSON representation using the normative @context `eopad-geojson.jsonld` defined in the previous section represents an EO Processing Application metadata as a [DCAT-2018] `dcat:DataService`:

```
"@type": "dcat:DataService"
```

Clients could use the media type `application/ld+json;profile="https://joinup.ec.europa.eu/node/154143/"` to obtain such representation if the Catalog server supports it. The Catalog server should then advertise this media type in its OSDD description and/or OpenAPI document. The `profile` parameter in the media type is defined in [RFC-Profile]. The media types `application/rdf+xml;profile="https://joinup.ec.europa.eu/node/154143/"` and `text/turtle;profile="https://joinup.ec.europa.eu/node/154143/"` could represent the equivalent RDF/XML and Turtle encodings.

An example of this encoding is given in section C.2.1 "JSON-LD (Expanded)".

In this case, it might be useful to also apply the alternative representation of geographical extent using the `dct:spatial` relation as presented in section 8.2 of [OGC17-084]. This allows to comply with the corresponding representation defined in [GeoDCAT-AP].

B.2.3. Relation with Schema.org

The JSON-LD representation obtained by expanding the GeoJSON representation using the alternative @context `eopad-geojson-sdo.jsonld` defined below represents an EO Processing Application metadata as a schema.org `schema:CreativeWork`:

```
"@type": "schema:CreativeWork"
```

Clients could use the media type `application/ld+json;profile="https://schema.org"` to obtain such representation if the Catalog server supports it. The Catalog server should then advertise this media type in its OSDD description and/or OpenAPI document.

An example of this encoding is given in section C.2.1 "JSON-LD (Expanded schema.org)". The advantage of this encoding is that it is understood as Structured Data by major search engines as shown in the figure below.

The screenshot shows the Google Structured Data Testing Tool interface. On the left, there is a code editor containing JSON-LD code. On the right, there is a results panel titled "CreativeWork". The results panel includes a summary table with columns for field names and their corresponding values. It also shows a detailed list of errors and warnings.

```

24  "@id": "http://fedeo.esa.int/services/MultiSensorNDVI",
25  "@type": "CreativeWork",
26  "additionalType": "http://inspire.ec.europa.eu/metadata-codelist/ResourceType
/service",
27  "alternateName": "MultiSensorNDVI",
28  "dateModified": "2019-02-01T00:00:00Z",
29  "description": "NDVI is calculated after the two bands values Near Infrared
and red. It is calculated by this formula : NDVI = (NIR-Red) / (NIR+Red)",
30  "hasPart": [
31    "http://geo.spacebel.be/opensearch
/request?parentIdentifier=EOP:ESA:FEDEO:COLLECTIONS&
uid=EOP%3AES%3ASentinel-1",
32    "http://geo.spacebel.be/opensearch
/request?parentIdentifier=EOP:ESA:FEDEO:COLLECTIONS&uid=EOP%3AES%3ASentinel-1"
33  ],
34  "inLanguage": "en",
35  "keywords": [
36    "Earth Online",
37    "ESA OADS",
38    "NDVI"
39  ],
40  "mentions": {
41    "@type": "OrganizationRole",
42    "member": {
43      "@type": "Organization",
44      "address": {
45        "addressCountry": "Italy",
46        "addressLocality": "Frascati",
47        "postalCode": "00044",
48        "streetAddress": "Via Galileo Galilei CP. 64"
49      },
50      "email": "eohelp@eo.esa.int",
51      "name": "ESA/ESRIN",
52      "telephone": "tel:39 06 94180777",
53      "url": "http://www.earth.esa.int"
54    },
55    "roleName": "originator"
56  },
57  "name": "Multi Sensor NDVI",
58  "potentialAction": [
59    {
60      "@type": "Action",
61      "alternateName": "http://www.opengis.net/spec/owc-geojson/1.0/req/docker
/image",
62      "subjectOf": {
63        "encodingFormat": "text/plain",
64        "text": "xxx/yyy/ZZZ:latest"
65      }
66    },
67  ],
68  "inLanguage": "en",
69  "keywords": [
70    "Earth Online",
71    "ESA OADS",
72    "NDVI"
73  ],
74  "name": "Multi Sensor NDVI",
75  "version": "1.0"
76}

```

CreativeWork	
ID:	http://fedeo.esa.int/services/MultiSensorNDVI
@type	CreativeWork
@id	http://fedeo.esa.int/services/MultiSensorNDVI
additionalType	/ResourceType/service
alternateName	MultiSensorNDVI
dateModified	2019-02-01T00:00:00Z
description	NDVI is calculated after the two bands values Near Infrared and red. It is calculated by this formula : NDVI = (NIR-Red) / (NIR+Red)
inLanguage	en
keywords	Earth Online
keyWords	ESA OADS
keyWords	NDVI
name	Multi Sensor NDVI
version	3.0
hasPart	
@type	CreativeWork
name	/request?parentIdentifier=EOP:ESA:FEDEO:COLLEC TIONS&uid=EOP%3AES%3ASentinel-1
mentions	
@type	OrganizationRole
roleName	originator
member	The property member is not recognized by Google for an object of type OrganizationRole.
@type	Organization
email	eohelp@eo.esa.int
name	ESA/ESRIN
telephone	tel:39 06 94180777

Figure 54. Google Structured Data Tool output

Below is the alternative @context `eopad-geojson-sdo.jsonld` used in this example.

`eopad-geojson-sdo.jsonld`

```
{
  "@context": {
    "@version": 1.1,
    "@vocab": "http://schema.org/",
    "id": "@id",
    "type": "@type",
    "Feature": "CreativeWork",
    "lang": "inLanguage",
    "versionInfo": "version",
    "kind": "additionalType",
    "page": "subjectOf",
    "Document": "WebPage",
    "links": "@nest",
    "profiles": {
      "@id": "accessibilityAPI",
      "@type": "@id",
      "@context": {
        "href": "@id"
      }
    },
    "profiles": null
}
```

```
"accessRights": null,  
"license": null,  
"subject": null,  
"theme": null,  
"versionNotes": null,  
"endpointDescription": null,  
"endpointURL": null,  
"properties": "@nest",  
"Offering": "Action",  
"operations": "target",  
"Operation": "EntryPoint",  
"offerings": {  
    "@id": "potentialAction",  
    "@context": {  
        "code": "identifier",  
        "content": "description"  
    }  
},  
"operations": {  
    "@id": "target",  
    "@context": {  
        "method": "httpMethod",  
        "href": "urlTemplate",  
        "type": "contentType",  
        "result": null  
    }  
},  
"contents": {  
    "@id": "subjectOf",  
    "@context": {  
        "type": "encodingFormat",  
        "content": "text"  
    }  
},  
"geometry": null,  
"doi": null,  
"previews": null,  
"bbox": null,  
"coordinates": null,  
"country-name": "addressCountry",  
"locality": "addressLocality",  
"contactPoint": "provider",  
"keyword": "keywords",  
"abstract": "description",  
"via": null,  
"data": "encoding",  
"alternates": "encoding",  
"href": "contentUrl",  
"role": "roleName",  
"Attribution": "OrganizationRole",  
"agent": {
```

```
    "@id": "member",
    "@type": "@id"
},
"qualifiedAttribution": "mentions",
"title": "name",
"identifier": "alternateName",
"Organization": "Organization",
"street-address": "streetAddress",
"postal-code": "postalCode",
"hasAddress": "address",
"email": "email",
"hasEmail": "email",
"hasName": "name",
"uri": "url",
"hasURL": "url",
"phone": "telephone",
"hasTelephone": "telephone",
"updated": "dateModified"
}
}
```

Appendix C: Encoding Examples

Extracts of the following examples were used in the body of the document. The current Annex includes the complete examples in the proposed encodings.

C.1. Service Interfaces

C.1.1. OpenSearch Description Document

C.1.2. OpenSearch Response

C.1.3. OpenAPI Description

```
{
  "openapi": "3.0.0",
  "info": {
    "description": "FedEO provides interoperable access, following ISO/OGC interface guidelines, to Earth Observation metadata.",
    "title": "FedEO Earth Observation Services Catalogue",
    "contact": {
      "name": "Spacebel s.a.",
      "url": "http://www.spacebel.be",
      "email": "yves.coene@spacebel.be"
    },
    "version": "1.0.0",
    "license": {
      "name": "Apache 2.0",
      "url": "http://www.apache.org/licenses/LICENSE-2.0.html"
    },
    "x-logo": {
      "url": "http://fedeo.esa.int/opensearch/images/esa_favicon.ico"
    },
    "x-providerName": "Copyright 2017-2019, European Space Agency.",
    "x-serviceName": "FedEO"
  },
  "servers": [
    {
      "url": "https://fedeo.esa.int"
    }
  ],
  "externalDocs": {
    "description": "Landing page for the API",
    "url": "http://fedeo.esa.int/opensearch/readme.html"
  },
  "x-@context": {
    "dc": "http://purl.org/dc/elements/1.1/",
    "eo": "http://a9.com/-/opensearch/extensions/eo/1.0/",
    "geo": "http://a9.com/-/opensearch/extensions/geo/1.0/"
  }
}
```

```
"os": "http://a9.com/-/spec/opensearch/1.1/",  
"semantic": "http://a9.com/-/opensearch/extensions/semantic/1.0/",  
"sru": "http://a9.com/-/opensearch/extensions/sru/2.0/",  
"time": "http://a9.com/-/opensearch/extensions/time/1.0/"  
},  
"paths": {  
    "/": {  
        "get": {  
            "tags": [  
                "LandingPage"  
            ],  
            "summary": "provide landing page (rel=\"self\") in OAPI Common format.",  
            "responses": {  
                "200": {  
                    "description": "landing page in OAPI Common encoding.",  
                    "content": {  
                        "application/json": {  
                            "schema": {  
                                "$ref": "#/components/schemas/root"  
                            }  
                        }  
                    }  
                }  
            }  
        }  
    },  
    "/conformance": {  
        "get": {  
            "tags": [  
                "Conformance"  
            ],  
            "summary": "provide conformance class information (rel=\"conformance\") in  
OAPI Common format.",  
            "responses": {  
                "200": {  
                    "description": "list of requirements classes in OAPI Common encoding.",  
                    "content": {  
                        "application/json": {  
                            "schema": {  
                                "$ref": "#/components/schemas/req-classes"  
                            }  
                        }  
                    }  
                }  
            }  
        },  
        "default": {  
            "description": "An error occurred.",  
            "content": {  
                "application/json": {  
                    "schema": {  
                        "$ref": "#/components/schemas/exception"  
                    }  
                }  
            }  
        }  
    }  
}
```

```

        }
    }
}
},
"/collections": {
    "get": {
        "tags": [
            "Collections"
        ],
        "summary": "Metadata about the resource collections",
        "description": "Metadata about the resource collections.",
        "responses": {
            "200": {
                "description": "metadata about the resource collections is successfully returned",
                "content": {
                    "application/json": {
                        "schema": {
                            "$ref": "#/components/schemas/collections"
                        },
                        "example": {
                            "collections": [
                                {
                                    "description": "Metadata records representing EO services and applications."
                                }
                            ],
                            "links": [
                                {
                                    "rel": "items",
                                    "href": "http://fedeo.esa.int/services",
                                    "type": "application/geo+json",
                                    "title": "Services and applications"
                                },
                                {
                                    "rel": "search",
                                    "href": "http://fedeo.esa.int/api",
                                    "type": "application/opensearchdescription+xml",
                                    "title": "Search interface"
                                },
                                {
                                    "rel": "describedBy",
                                    "href": "http://schemas.opengis.net/eopad-geojson/1.0/eopad-geojson-schema.json",
                                    "type": "application/schema+json",
                                    "title": "JSON schema for items belonging to this collection"
                                }
                            ],
                            "id": "services",
                            "title": "EO services and applications"
                        }
                    }
                }
            }
        }
    }
}

```

```

        }
    ],
    "links": [
        {
            "rel": "self",
            "href": "http://fedeo.esa.int/eo-catalog/collections",
            "type": "application/json",
            "title": "this document"
        }
    ]
}
},
"500": {
    "description": "Unexpected error"
}
}
},
"/collections/{collection-id)": {
    "get": {
        "tags": [
            "Collections"
        ],
        "summary": "Metadata about a resource collection",
        "description": "Metadata about a resource collection.",
        "responses": {
            "200": {
                "description": "metadata about a resource collection is successfully returned",
                "content": {
                    "application/json": {
                        "schema": {
                            "$ref": "#/components/schemas/collection"
                        },
                        "example": {
                            "description": "Metadata records representing EO services and applications."
                        },
                        "links": [
                            {
                                "rel": "items",
                                "href": "http://fedeo.esa.int/eo-catalog/services",
                                "type": "application/geo+json",
                                "title": "Services and applications"
                            }
                        ],
                        "id": "services",
                        "title": "EO services and applications"
                    }
                }
            }
        }
    }
}
}

```

```

    }
},
"500": {
    "description": "Unexpected error"
}
},
"parameters": [
{
    "schema": {
        "default": "services",
        "type": "string",
        "enum": [
            "services"
        ]
    },
    "in": "path",
    "name": "collection-id",
    "description": "Collection identifier.",
    "required": true
}
]
}
},
"/api": {
    "get": {
        "tags": [
            "APIDescription"
        ],
        "summary": "describe the catalogue interface (rel=\"service\")",
        "description": "The endpoint returns meta information about the interface as an OpenSearch Description Document (OSDD) or OpenAPI 3.0 Document in YAML or JSON format.\n",
        "responses": {
            "200": {
                "description": "interface description in requested format",
                "content": {
                    "application/opensearchdescription+xml": {},
                    "application/openapi+json;version=3.0": {},
                    "application/json": {}
                }
            },
            "415": {
                "description": "Unsupported Media Type."
            },
            "500": {
                "description": "Unexpected error"
            }
        }
    }
},
"/services": {

```

```

"get": {
  "tags": [
    "Services"
  ],
  "summary": "service or application search (rel=\"data\")",
  "description": "The endpoint returns information about the *FedEO* service or application matching specific filtering criteria such as organisation, platform, instrument, title, keyword and lists the services or applications.\n",
  "parameters": [
    {
      "name": "maximumRecords",
      "in": "query",
      "x-value": "{os:count}",
      "description": "Number of records {os:count}.",
      "required": false,
      "schema": {
        "type": "integer",
        "format": "int32",
        "default": 10,
        "minimum": 0,
        "exclusiveMinimum": false,
        "maximum": 50,
        "exclusiveMaximum": false
      }
    },
    {
      "name": "startRecord",
      "in": "query",
      "x-value": "{os:startIndex}",
      "description": "Start index of first result {os:startIndex}.",
      "required": false,
      "schema": {
        "type": "integer",
        "format": "int32",
        "default": 1,
        "minimum": 1
      }
    },
    {
      "name": "query",
      "in": "query",
      "x-value": "{os:searchTerms}",
      "description": "Free text search term {os:searchTerms}.",
      "required": false,
      "schema": {
        "type": "string"
      }
    },
    {
      "name": "parentIdentifier",
      "x-value": "{eo:parentIdentifier}",

```

```

    "in": "query",
    "description": "Identifier of subcatalogue {eo:parentIdentifier}.",
    "required": false,
    "schema": {
        "type": "string",
        "enum": [
            "EOP:ESA:FEDEO",
            "EOP:NASA:CMR",
            "EOP:VITO:PDF",
            "EOP:JAXA:CATS-I",
            "EOP:DLR:GEOSERVICE",
            "EOP:ESA:FEDEO:COLLECTIONS",
            "EOP:EUMETSAT"
        ]
    }
},
{
    "name": "organisationName",
    "in": "query",
    "x-value": "{eo:organisationName}",
    "description": "Name of data provider {eo:organisationName}.",
    "required": false,
    "schema": {
        "type": "string",
        "enum": [
            "ESA",
            "VITO",
            "JAXA",
            "DLR"
        ]
    }
},
{
    "name": "platform",
    "in": "query",
    "x-value": "{eo:platform}",
    "description": "Satellite name {eo:platform}.",
    "required": false,
    "schema": {
        "type": "string"
    }
},
{
    "name": "instrument",
    "in": "query",
    "x-value": "{eo:instrument}",
    "description": "Instrument name.",
    "required": false,
    "schema": {
        "type": "string"
    }
}

```

```

},
{
  "name": "startDate",
  "in": "query",
  "x-value": "{time:start}",
  "description": "Start datetime of temporal constraint {time:start}.",
  "required": false,
  "allowEmptyValue": true,
  "schema": {
    "type": "string",
    "format": "date-time"
  }
},
{
  "name": "endDate",
  "in": "query",
  "x-value": "{time:end}",
  "description": "End datetime of temporal constraint {time:end}.",
  "required": false,
  "allowEmptyValue": true,
  "schema": {
    "type": "string",
    "format": "date-time"
  }
},
{
  "name": "bbox",
  "in": "query",
  "x-value": "{geo:box}",
  "description": "Area of interest {geo:box}.",
  "required": false,
  "schema": {
    "type": "string"
  }
},
{
  "name": "uid",
  "in": "query",
  "x-value": "{geo:uid}",
  "description": "Local identifier {geo:uid} of the service or application returned as \\"dc:identifier\\" in response.",
  "required": false,
  "schema": {
    "type": "string"
  }
},
{
  "name": "subject",
  "in": "query",
  "x-value": "{eo:keyword}",
  "x-example": "ESA OADS",

```

```

    "description": "Keyword appearing in metadata record {eo:keyword}.",
    "required": false,
    "schema": {
        "type": "string"
    }
},
{
    "name": "title",
    "in": "query",
    "x-value": "{eo:title}",
    "description": "Metadata record title {eo:title}.",
    "required": false,
    "schema": {
        "type": "string"
    }
},
{
    "name": "classifiedAs",
    "in": "query",
    "x-value": "{semantic:classifiedAs?}",
    "description": "Keyword URI appearing in metadata record {semantic:classifiedAs?}.",
    "required": false,
    "schema": {
        "type": "string"
    }
},
{
    "name": "facetLimit",
    "in": "query",
    "x-value": "{sru:facetLimit?}",
    "description": "Maximum number of counts for a facet in faceted results {sru:facetLimit?}.",
    "required": false,
    "schema": {
        "type": "string"
    }
}
],
"responses": {
    "200": {
        "description": "An array of services or applications",
        "content": {
            "application/geo+json": {
                "schema": {
                    "$ref": "#/components/schemas/FeatureCollection"
                },
                "example": {}
            }
        }
    }
},

```

```

    "400": {
      "description": "Bad request (invalid parameter)"
    },
    "415": {
      "description": "Unsupported Media Type."
    },
    "500": {
      "description": "Unexpected error"
    }
  }
},
"post": {
  "tags": [
    "Services"
  ],
  "description": "Insert a new service or application to the catalogue.",
  "requestBody": {
    "description": "Insert a new service or application to the catalogue.",
    "required": true,
    "content": {
      "application/geo+json": {
        "schema": {
          "$ref": "#/components/schemas/Feature"
        }
      }
    }
  },
  "responses": {
    "201": {
      "description": "Service or application has been successfully inserted"
    },
    "401": {
      "description": "Unauthorized"
    },
    "409": {
      "description": "Conflict - resource already exists"
    },
    "415": {
      "description": "Unsupported Media Type."
    },
    "500": {
      "description": "Unexpected error"
    }
  }
},
"/services/{service-id}": {
  "get": {
    "description": "Search a service or application from the catalogue.",
    "tags": [
      "Service"
    ]
  }
}

```

```

],
"parameters": [
{
  "name": "service-id",
  "x-value": "{geo:uid}",
  "in": "path",
  "required": true,
  "description": "service or application identifier.",
  "schema": {
    "type": "string"
  }
}
],
"responses": {
  "200": {
    "description": "An array with requested service or application metadata",
    "content": {
      "application/geo+json": {
        "schema": {
          "$ref": "#/components/schemas/FeatureCollection"
        },
        "example": {}
      }
    }
  },
  "400": {
    "description": "Bad request (invalid parameter)"
  },
  "404": {
    "description": "service or application not found"
  },
  "500": {
    "description": "Unexpected error"
  }
},
"put": {
  "tags": [
    "Service"
  ],
  "description": "Update a service record in the catalogue.",
  "parameters": [
    {
      "name": "service-id",
      "in": "path",
      "required": true,
      "description": "service identifier.",
      "schema": {
        "type": "string"
      }
    }
  ]
}
]

```

```

],
"requestBody": {
  "description": "Update a service record in the catalogue.",
  "required": true,
  "content": {
    "application/geo+json": {
      "schema": {
        "$ref": "#/components/schemas/Feature"
      }
    }
  }
},
"responses": {
  "200": {
    "description": "Service has been successfully updated."
  },
  "400": {
    "description": "Bad request."
  },
  "404": {
    "description": "Not found."
  },
  "415": {
    "description": "Unsupported Media Type."
  },
  "500": {
    "description": "Unexpected error"
  }
},
"delete": {
  "tags": [
    "Service"
  ],
  "description": "Delete a service or application from the catalogue.",
  "parameters": [
    {
      "name": "service-id",
      "in": "path",
      "required": true,
      "description": "service or application identifier.",
      "schema": {
        "type": "string"
      }
    }
  ],
  "responses": {
    "204": {
      "description": "Service has been deleted."
    },
    "404": {

```

```

        "description": "Not found."
    },
    "500": {
        "description": "Unexpected error"
    }
}
}
},
"components": {
    "schemas": {
        "FeatureCollection": {
            "type": "object"
        },
        "Feature": {
            "type": "object"
        },
        "exception": {
            "type": "object",
            "required": [
                "code"
            ],
            "properties": {
                "code": {
                    "type": "string"
                },
                "description": {
                    "type": "string"
                }
            }
        },
        "root": {
            "type": "object",
            "required": [
                "links"
            ],
            "properties": {
                "links": {
                    "type": "array",
                    "items": {
                        "$ref": "#/components/schemas/link"
                    }
                }
            }
        },
        "collections": {
            "type": "array",
            "items": {
                "$ref": "#/components/schemas/collection"
            }
        },
    }
},

```

```

"collection": {
  "type": "object",
  "required": [
    "id",
    "links"
  ],
  "properties": {
    "extent": {
      "$ref": "#/components/schemas/extent"
    },
    "crs": {
      "default": [
        "http://www.opengis.net/def/crs/OGC/1.3/CRS84"
      ],
      "description": "the list of coordinate reference systems supported by the API; the first item is the default coordinate reference system",
      "type": "array",
      "items": {
        "type": "string"
      },
      "example": [
        "http://www.opengis.net/def/crs/OGC/1.3/CRS84",
        "http://www.opengis.net/def/crs/EPSG/0/4326"
      ]
    },
    "description": {
      "type": "string"
    },
    "links": {
      "type": "array",
      "items": {
        "$ref": "#/components/schemas/link"
      }
    },
    "id": {
      "type": "string"
    },
    "title": {
      "type": "string"
    }
  }
},
"extent": {
  "type": "object",
  "properties": {
    "trs": {
      "default": "http://www.opengis.net/def/uom/ISO-8601/0/Gregorian",
      "description": "Temporal reference system of the coordinates in the temporal extent (property `temporal`). In the Core, only the Gregorian calendar is supported. Extensions may support additional temporal reference systems.",
      "type": "string",
    }
  }
}

```

```

"enum": [
    "http://www.opengis.net/def/uom/ISO-8601/0/Gregorian"
]
},
"crs": {
    "default": "http://www.opengis.net/def/crs/OGC/1.3/CRS84",
    "description": "Coordinate reference system of the coordinates in the spatial extent (property `spatial`). In the Core, only WGS84 longitude/latitude is supported. Extensions may support additional coordinate reference systems.",
    "type": "string",
    "enum": [
        "http://www.opengis.net/def/crs/OGC/1.3/CRS84"
    ]
},
"spatial": {
    "minItems": 4,
    "maxItems": 4,
    "description": "West, south, east, north edges of the spatial extent. The minimum and maximum values apply to the coordinate reference system WGS84 longitude/latitude that is supported in the Core. If, for example, a projected coordinate reference system is used, the minimum and maximum values need to be adjusted.",
    "type": "array",
    "items": {
        "maximum": 180,
        "type": "number",
        "minimum": -180
    },
    "example": [
        -180,
        -90,
        180,
        90
    ]
},
"temporal": {
    "minItems": 2,
    "maxItems": 2,
    "description": "Begin and end times of the temporal extent.",
    "type": "array",
    "items": {
        "format": "dateTime",
        "type": "string"
    },
    "example": [
        "2011-11-11T12:22:11Z",
        "2012-11-24T12:32:43Z"
    ]
}
}
},

```

```
"req-classes": {
  "type": "object",
  "required": [
    "conformsTo"
  ],
  "properties": {
    "conformsTo": {
      "type": "array",
      "items": {
        "type": "string"
      },
      "example": [
        "http://www.opengis.net/spec/eopad-geojson/1.0/req/core",
        "http://www.opengis.net/spec/os-geojson/1.0/req/core",
        "http://www.opengis.net/spec/owc-geojson/1.0/req/core"
      ]
    }
  }
},
"link": {
  "type": "object",
  "required": [
    "href"
  ],
  "properties": {
    "href": {
      "type": "string"
    },
    "rel": {
      "type": "string",
      "example": "prev"
    },
    "type": {
      "type": "string",
      "example": "application/geo+json"
    },
    "hreflang": {
      "type": "string",
      "example": "en"
    },
    "title": {
      "type": "string",
      "example": "previous page"
    }
  }
}
}
```

C.2. Metadata Examples

C.2.1. Example 1: MultiSensorNDVI Application

GeoJSON

```
{
  "id": "http://fedeo.esa.int/services/MultiSensorNDVI",
  "type": "Feature",
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [
          [
            [
              [
                [
                  [
                    [
                      [
                        [
                          [
                            [
                              [
                                [
                                  [
                                    [
                                      [
                                        [
                                          [
                                            [
                                              [
                                                [
                                                  [
                                                    [
                                                      [
                                                        [
                                                          [
                                                            [
                                                              [
                                                                [
                                                                  [
                                                                    [
                                                                      [
                                                                        [
                                                                          [
                                                                            [
                                                                              [
                                                                                [
                                                                                  [
                                                                                    [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
                                                                                      [
................................................................
```

```

        "prefLabel": "NDVI"
    }
],
"keyword": [
    "Earth Online",
    "ESA OADS",
    "NDVI"
],
"links": {
    "profiles": [
        {
            "href": "http://www.opengis.net/spec/owc-geojson/1.0/req/core"
        },
        {
            "href": "http://www.opengis.net/spec/eopad-geojson/1.0/req/core"
        }
    ]
},
"hasPart": [
    "http://geo.spacebel.be/opensearch/request?parentIdentifier=EOP:ESA:FEDEO:COLLECTIONS&uid=EOP%3AES%3ASentinel-1",
    "http://geo.spacebel.be/opensearch/request?parentIdentifier=EOP:ESA:FEDEO:COLLECTIONS&uid=EOP%3AES%3ASentinel-2"
],
"endpointDescription": [
    "http://tep.esa.int/wps/processing?REQUEST=GetCapabilities&SERVICE=WPS"
],
"endpointURL":
"http://tep.esa.int/wps/processing?REQUEST=GetCapabilities&SERVICE=WPS",
"offerings": [
    {
        "type": "Offering",
        "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/docker/image"
    },
    {
        "contents": [
            {
                "type": "text/plain",
                "content": "xxx/yyy/ZZZ:latest"
            }
        ]
    },
    {
        "type": "Offering",
        "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wps",
        "operations": [
            {
                "code": "GetCapabilities",
                "method": "GET",
                "type": "application/xml",
            }
        ]
    }
]
}

```

```

        "href": "http://tep.esa.int/wps/processing?REQUEST=GetCapabilities&SERVICE=WPS",
        "result": {
            "@type": "Content",
            "type": "application/xml",
            "content": "<wps:Capabilities>...</wps:Capabilities>"
        }
    }
],
"subject": [
{
    "term": "http://inspire.ec.europa.eu/metadata-
codelist/TopicCategory/imageryBaseMapsEarthCover"
}
],
"license": [
{
    "type": "LicenseDocument",
    "label": "Fast Registration with immediate access are available online
upon fast registration"
}
],
"accessRights": [
{
    "type": "RightsStatement",
    "label": "Utilisation of this data is subject to the Terms and
Conditions for ESA's Third Party Missions scheme"
}
],
"page": [
{
    "id": "https://earth.esa.int/web/guest/-/multi-sensor-ndvi.html",
    "type": "Document"
}
],
"wasUsedBy": [
{
    "type": "Activity",
    "generated": {
        "type": "Entity",
        "degree": "http://inspire.ec.europa.eu/metadata-
codelist/DegreeOfConformity/conformant",
        "description": "See the referenced specification"
    },
    "qualifiedAssociation": {
        "type": "Association",
        "hadPlan": {
            "type": "Plan",
            "wasDerivedFrom": {

```

```

        "type": "Standard",
        "title": "COMMISSION REGULATION (EU) No 1089/2010 of 23
November 2010 implementing Directive 2007/2/EC of the European Parliament and of the
Council as regards interoperability of spatial data sets and services",
        "issued": "2010-12-08T00:00:00Z"
    }
}
},
],
"contactPoint": [
{
    "type": "Organization",
    "email": "mailto:eohelp@eo.esa.int",
    "name": "ESA/ESRIN",
    "phone": "tel:+39 06 94180777",
    "uri": "http://www.earth.esa.int",
    "hasAddress": {
        "country-name": "Italy",
        "postal-code": "00044",
        "locality": "Frascati",
        "street-address": "Via Galileo Galilei CP. 64"
    }
}
],
"qualifiedAttribution": [
{
    "type": "Attribution",
    "agent": [
{
        "type": "Organization",
        "email": "eohelp@eo.esa.int",
        "name": "ESA/ESRIN",
        "phone": "tel:39 06 94180777",
        "uri": "http://www.earth.esa.int",
        "hasAddress": {
            "country-name": "Italy",
            "postal-code": "00044",
            "locality": "Frascati",
            "street-address": "Via Galileo Galilei CP. 64"
        }
    }
]
},
{
    "role": "originator"
}
]
}
}

```

JSON-LD (Compacted)

```
{  
  "@context": "http://schemas.opengis.net/eopad-geojson/1.0/eopad-geojson.jsonld",  
  "type": "Feature",  
  ... same content as example in previous section ...  
}
```

JSON-LD (Expanded)

```
{  
  "@context": {  
    "@version": 1.1,  
    "dct": "http://purl.org/dc/terms/",  
    "atom": "http://www.w3.org/2005/Atom/",  
    "iana": "http://www.iana.org/assignments/relation/",  
    "os": "http://a9.com/-/spec/opensearch/1.1/",  
    "eop": "http://www.opengis.net/ont/eo-geojson/1.0/",  
    "owc": "http://www.opengis.net/ont/owc/1.0/",  
    "gj": "https://purl.org/geojson/vocab#",  
    "gsp": "http://www.opengis.net/ont/geosparql#",  
    "vcard": "http://www.w3.org/2006/vcard/ns#",  
    "skos": "http://www.w3.org/2004/02/skos/core#",  
    "dcat": "http://www.w3.org/ns/dcat#",  
    "xsd": "http://www.w3.org/2001/XMLSchema#",  
    "prov": "http://www.w3.org/ns/prov#",  
    "locn": "http://www.w3.org/ns/locn#",  
    "foaf": "http://xmlns.com/foaf/0.1/",  
    "schema": "http://schema.org/",  
    "rdfs": "http://www.w3.org/2000/01/rdf-schema#",  
    "adms": "http://www.w3.org/ns/adms#",  
    "owl": "http://www.w3.org/2002/07/owl#",  
    "wps": "http://www.opengis.net/wps/2.0/",  
    "ldp": "http://www.w3.org/ns/ldp#"  
  },  
  "@id": "http://fedeo.esa.int/services/MultiSensorNDVI",  
  "@type": "dcat:DataService",  
  "dct:accessRights": {  
    "@type": "dct:RightsStatement",  
    "rdfs:label": "Utilisation of this data is subject to the Terms and Conditions  
for ESA's Third Party Missions scheme"  
  },  
  "dct:description": "NDVI is calculated after the two bands values Near Infrared  
and red. It is calculated by this formula : NDVI = (NIR-Red)/(NIR+Red)",  
  "dct:hasPart": [  
  ]  
}
```

```

"@id": "http://geo.spacebel.be/opensearch/request?parentIdentifier=EOP:ESA:FEDEO:COLLECTIONS&uid=EOP%3AES%3ASentinel-1"
},
{
  "@id": "http://geo.spacebel.be/opensearch/request?parentIdentifier=EOP:ESA:FEDEO:COLLECTIONS&uid=EOP%3AES%3ASentinel-2"
}
],
"dct:identifier": "MultiSensorNDVI",
"dct:language": {
  "@id": "http://id.loc.gov/vocabulary/iso639-1/en"
},
"dct:license": {
  "@type": "dct:LicenseDocument",
  "rdfs:label": "Fast Registration with immediate access are available online upon fast registration"
},
"dct:modified": "2019-06-19T00:00:00Z",
"dct:subject": {
  "@id": "http://inspire.ec.europa.eu/metadata-codelist/TopicCategory/imageryBaseMapsEarthCover"
},
"dct:title": "Multi Sensor NDVI",
"dct:type": {
  "@id": "http://inspire.ec.europa.eu/metadata-codelist/ResourceType/service"
},
"owc:links": {
  "iana:profile": [
    {
      "@id": "http://www.opengis.net/spec/owc-geojson/1.0/req/core"
    },
    {
      "@id": "http://www.opengis.net/spec/eopad-geojson/1.0/req/core"
    }
  ]
},
"owl:versionInfo": "3.0",
"adms:versionNotes": "Updated for use in OGC Testbed-15.",
"dcat:contactPoint": {
  "@type": "vcard:Organization",
  "vcard:hasAddress": {
    "vcard:country-name": "Italy",
    "vcard:locality": "Frascati",
    "vcard:postal-code": "00044",
    "vcard:street-address": "Via Galileo Galilei CP. 64"
  },
  "vcard:hasEmail": "mailto:eohelp@eo.esa.int",
  "vcard:hasName": "ESA/ESRIN",
  "vcard:hasTelephone": "tel:+39 06 94180777",
}

```

```
"vcard:hasURL": "http://www.earth.esa.int"
},
"dcat:endpointDescription": [
{
  "@id":
"http://tep.esa.int/wps/processing?REQUEST=GetCapabilities&SERVICE=WPS"
},
{
  "@type": "owc:Offering",
  "owc:code": "http://www.opengis.net/spec/owc-geojson/1.0/req/docker/image"
},
{
  "owc:contents": {
    "owc:content": "xxx/yyy/ZZZ:latest",
    "owc:type": "text/plain"
  }
},
{
  "@type": "owc:Offering",
  "owc:code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wps",
  "owc:operations": {
    "owc:code": "GetCapabilities",
    "owc:href":
"http://tep.esa.int/wps/processing?REQUEST=GetCapabilities&SERVICE=WPS",
    "owc:method": "GET",
    "owc:result": {
      "@type": "owc:Content",
      "owc:content": "<wps:Capabilities>...</wps:Capabilities>",
      "owc:type": "application/xml"
    },
    "owc:type": "application/xml"
  }
}
],
  "dcat:keyword": [
    "Earth Online",
    "ESA OADS",
    "NDVI"
  ],
  "prov:qualifiedAttribution": {
    "@type": "prov:Attribution",
    "dct:type": {
      "@id": "http://inspire.ec.europa.eu/metadata-codelist/ResponsiblePartyRole/originator"
    },
    "prov:agent": {
      "@type": "vcard:Organization",
      "vcard:hasAddress": {
        "vcard:country-name": "Italy",
        "vcard:locality": "Frascati",
        "vcard:postal-code": "00044",
        "vcard:street-address": "Via Galileo Galilei CP. 64"
      }
    }
  }
]
```

```

        },
        "vcard:hasEmail": "eohelp@eo.esa.int",
        "vcard:hasName": "ESA/ESRIN",
        "vcard:hasTelephone": "tel:39 06 94180777",
        "vcard:hasURL": "http://www.earth.esa.int"
    }
},
"prov:wasUsedBy": {
    "@type": "prov:Activity",
    "prov:generated": {
        "@type": "prov:Entity",
        "dct:type": {
            "@id": "http://inspire.ec.europa.eu/metadata-
codeList/DegreeOfConformity/conformant"
        }
    },
    "prov:qualifiedAssociation": {
        "@type": "prov:Association",
        "prov:hadPlan": {
            "@type": "prov:Plan",
            "prov:wasDerivedFrom": {
                "@type": "dct:Standard",
                "dct:issued": "2010-12-08",
                "dct:title": "COMMISSION REGULATION (EU) No 1089/2010 of 23
November 2010 implementing Directive 2007/2/EC of the European Parliament and of the
Council as regards interoperability of spatial data sets and services"
            }
        }
    }
},
"foaf:page": {
    "@id": "https://earth.esa.int/web/guest/-/multi-sensor-ndvi.html",
    "@type": "foaf:Document"
},
"gj:geometry": {
    "@type": "gj:Polygon",
    "gj:coordinates": [
        -180,
        -90,
        180,
        -90,
        180,
        90,
        -180,
        90,
        -180,
        -90
    ]
}
}

```

JSON-LD (Expanded - schema.org)

This section shows the result of applying an alternative @context eopad-geojson-sdo.jsonld to the same GeoJSON encoding. It uses only schema.org classes and properties, does not represent all properties available in the original GeoJSON encoding but facilitates discovery of key information by mass market search engines.

```
{
  "@context": {
    "@version": 1.1,
    "@vocab": "http://schema.org/"
  },
  "@id": "http://fedeo.esa.int/services/MultiSensorNDVI",
  "@type": "CreativeWork",
  "additionalType": "http://inspire.ec.europa.eu/metadata-
  codeList/ResourceType/service",
  "alternateName": "MultiSensorNDVI",
  "dateModified": "2019-02-01T00:00:00Z",
  "description": "NDVI is calculated after the two bands values Near Infrared and red. It is calculated by this formula : NDVI = (NIR-Red)/(NIR+Red)",
  "inLanguage": "en",
  "keywords": [
    "Earth Online",
    "ESA OADS",
    "NDVI"
  ],
  "mentions": {
    "@type": "OrganizationRole",
    "member": {
      "@type": "Organization",
      "address": {
        "addressCountry": "Italy",
        "addressLocality": "Frascati",
        "postalCode": "00044",
        "streetAddress": "Via Galileo Galilei CP. 64"
      },
      "email": "eohelp@eo.esa.int",
      "name": "ESA/ESRIN",
      "telephone": "tel:39 06 94180777",
      "url": "http://www.earth.esa.int"
    },
    "roleName": "originator"
  },
  "name": "Multi Sensor NDVI",
  "potentialAction": [
    {
      "@type": "Action",
      "identifier": "http://www.opengis.net/spec/owc-
      geojson/1.0/req/docker/image",
      "subjectOf": {
        "@type": "Image"
      }
    }
  ]
}
```

```

        "encodingFormat": "text/plain",
        "text": "xxx/yyy/ZZZ:latest"
    }
},
{
    "@type": "Action",
    "identifier": "http://www.opengis.net/spec/owc-geojson/1.0/req/wps",
    "target": {
        "contentType": "application/xml",
        "httpMethod": "GET",
        "identifier": "GetCapabilities",
        "urlTemplate":
"http://tep.esa.int/wps/processing?REQUEST=GetCapabilities&SERVICE=WPS"
    }
}
],
"provider": {
    "@type": "Organization",
    "address": {
        "addressCountry": "Italy",
        "addressLocality": "Frascati",
        "postalCode": "00044",
        "streetAddress": "Via Galileo Galilei CP. 64"
    },
    "email": "mailto:eohelp@eo.esa.int",
    "name": "ESA/ESRIN",
    "telephone": "tel:+39 06 94180777",
    "url": "http://www.earth.esa.int"
},
"subjectOf": {
    "@id": "https://earth.esa.int/web/guest/-/multi-sensor-ndvi.html",
    "@type": "WebPage"
},
"version": "3.0"
}

```

C.2.2. Example 2: NDVI Calculation

Application Package (DeployProcess)

```
{
"processDescription": {
    "process": {
        "id": "org.n52.project.tb15.eopad.NdviCalculation",
        "title": "Calculation of NDVI using the SNAP toolbox for Sentinel-2",
        "description": "Calculation of NDVI using the SNAP toolbox for Sentinel-2",
        "version": "1.0.0",
        "keywords": ["Vegetation", "NDVI"],
        "inputs": [
            {

```

```

    "id": "source",
    "title": "Source Input Image (sentinel-2 product in SAFE format)",
    "description": "Source Input Image (sentinel-2 product in SAFE format).
Either provided inline, by reference or by product ID",
    "minOccurs": 1,
    "maxOccurs": 1,
    "input": {
        "formats": [
            {
                "mimeType": "text/plain",
                "default": true
            },
            {
                "mimeType": "application/zip",
                "default": false
            }
        ]
    }
},
{
    "id": "red_source_band",
    "title": "the band to be used for the red band",
    "minOccurs": 0,
    "maxOccurs": 1,
    "input": {
        "literalDataDomains": [
            {
                "default": true,
                "dataType": { "name": "integer" },
                "valueDefinition": [
                    {
                        "minimumValue": "1",
                        "maximumValue": "13",
                        "rangeClosure": "closed"
                    }
                ]
            }
        ]
    }
},
{
    "id": "nir_source_band",
    "title": "the band to be used for the near-infrared band",
    "minOccurs": 0,
    "maxOccurs": 1,
    "input": {
        "literalDataDomains": [
            {
                "default": true,
                "dataType": { "name": "integer" },
                "valueDefinition": [

```

```

        {
            "minimumValue": "1",
            "maximumValue": "13",
            "rangeClosure": "closed"
        }
    ]
}
]
},
{
    "id": "red_factor",
    "title": "the factor of the NRI",
    "minOccurs": 0,
    "maxOccurs": 1,
    "input": {
        "literalDataDomains": [
            {
                "default": true,
                "defaultValue": "1.0",
                "dataType": { "name": "float" },
                "valueDefinition": [
                    {
                        "minimumValue": "0.0",
                        "rangeClosure": "closed"
                    }
                ]
            }
        ]
    }
},
{
    "id": "nir_factor",
    "title": "the factor of the NIR",
    "minOccurs": 0,
    "maxOccurs": 1,
    "input": {
        "literalDataDomains": [
            {
                "default": true,
                "defaultValue": "1.0",
                "dataType": { "name": "float" },
                "valueDefinition": [
                    {
                        "minimumValue": "0.0",
                        "rangeClosure": "closed"
                    }
                ]
            }
        ]
    }
}

```

```

},
{
  "id": "cloud_coverage",
  "title": "The cloud coverage of the scene.",
  "minOccurs": 0,
  "maxOccurs": 1,
  "input": {
    "literalDataDomains": [
      {
        "default": true,
        "dataType": { "name": "float" },
        "valueDefinition": [
          {
            "minimumValue": "0.0",
            "maximumValue": "100.0",
            "rangeClosure": "closed"
          }
        ]
      }
    ]
  }
},
{
  "id": "max_cloud_coverage",
  "title": "The maximum cloud coverage of the scene.",
  "minOccurs": 0,
  "maxOccurs": 1,
  "input": {
    "literalDataDomains": [
      {
        "default": true,
        "dataType": { "name": "float" },
        "valueDefinition": [
          {
            "minimumValue": "0.0",
            "maximumValue": "100.0",
            "rangeClosure": "closed"
          }
        ]
      }
    ]
  }
},
{
  "outputs": [
    {
      "id": "raster",
      "title": "The raster of the resulting NDVI calculation",
      "output": {
        "formats": [
          {

```

```

        "mimeType": "image/geotiff",
        "default": true
    }
]
}
]
},
"jobControlOptions": ["async-execute"],
"outputTransmission": ["reference"]
},
"immediateDeployment": true,
"executionUnit": [
{
"unit": {
"type": "docker",
"image": "52north/testbed-eopad-ndvi:latest"
}
}
],
"deploymentProfileName": "http://www.opengis.net/profiles/eoc/dockerizedApplication"
}

```

GeoJSON

```
{
  "id": "http://fedeo.esa.int/services/NdviCalculation",
  "type": "Feature",
  "geometry": null,
  "properties": {
    "identifier": "NdviCalculation",
    "kind": "http://inspire.ec.europa.eu/metadata-codelist/ResourceType/service",
    "title": "Calculation of NDVI using the SNAP toolbox for Sentinel-2",
    "abstract": "Calculation of NDVI using the SNAP toolbox for Sentinel-2",
    "updated": "2019-05-20T00:00:00Z",
    "versionInfo": "0.1",
    "versionNotes": "Initial version of application for use in OGC Testbed-15.",
    "lang": "en",
    "keyword": [
      "Vegetation"
    ],
    "links": {
      "profiles": [
        {
          "href": "http://www.opengis.net/spec/owc-geojson/1.0/req/core"
        },
        {
          "href": "http://www.opengis.net/spec/eopad-geojson/1.0/req/core"
        }
      ]
    }
  }
}
```

```

"alternates": [
    {
        "href": "http://fedeo.esa.int/services/NdviCalculation?httpAccept=application/vnd.iso.19139-2%2Bxml",
        "type": "application/vnd.iso.19139-2+xml",
        "title": "ISO 19139-2 metadata"
    }
],
"describedby": [
    {
        "href": "http://fedeo.esa.int/services/NdviCalculation?httpAccept=text/html",
        "type": "text/html",
        "title": "Described by"
    }
],
"offerings": [
    {
        "type": "Offering",
        "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/docker/image",
        "contents": [
            {
                "type": "text/plain",
                "content": "docker.52north.org/eopad/snap-sentinel-2-ndvi:latest"
            }
        ]
    },
    {
        "type": "Offering",
        "code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wps",
        "operations": [
            {
                "code": "DescribeProcess",
                "method": "GET",
                "type": "application/json",
                "href": "http://52north.org/eopad/wps/processes/NdviCalculation",
                "result": {
                    "type": "application/json",
                    "content": {
                        "processDescription": {
                            "process": {
                                "id": "NdviCalculation",
                                "title": "Calculation of NDVI using the SNAP toolbox for Sentinel-2",
                                "abstract": "Calculation of NDVI using the SNAP toolbox for Sentinel-2",
                            }
                        }
                    }
                }
            }
        ]
    }
]

```

```

    "keywords": [
        "Vegetation"
    ],
    "inputs": [
        {
            "id": "source",
            "title": "Source Input Image
(sentinel-2 product in SAFE format)",
            "input": {
                "formats": [
                    {
                        "mimeType":
"application/zip",
                        "default": true
                    }
                ],
                "minOccurs": "1",
                "maxOccurs": "1",
                "additionalParameters": [
                    {
                        "role":
"http://www.opengis.net/eoc/applicationContext/inputMetadata",
                        "parameters": [
                            {
                                "_comment": "The
Profile element allows to define a more specific format than the registered MIME Media
Type (e.g. GeoTiff)",
                                "name": "Profile",
                                "values": [
                                    "http://www.opengis.net/eoc/formats/sensor/Sentinel-2"
                                ]
                            }
                        ]
                    },
                    {
                        "role":
"http://www.opengis.net/eoc/applicationContext/inputConstraints",
                        "parameters": [
                            {
                                "name": "CollectionsConstraints",
                                "values": [
                                    "Sentinel-2"
                                ]
                            },
                            {
                                "name": "SpatialConstraints",
                                "values": [

```

```
"TimeConstraints",
  "values": [
    "2019-05-01",
    "2019-12-31"
  ]
}
]
}
],
{
  "id": "red_source_band",
  "title": "the band to be used for the red band",
  "input": {
    "literalDataDomains": [
      {
        "valueDefinition": {
          "anyValue": true
        },
        "dataType": {
          "name": "number"
        }
      }
    ],
    "minOccurs": "1",
    "maxOccurs": "1"
  },
  {
    "id": "nir_source_band",
    "title": "the band to be used for the near-infrared band",
    "input": {
      "literalDataDomains": [
        {
          "valueDefinition": {
            "anyValue": true
          },
          "dataType": {
            "name": "number"
          }
        }
      ]
    }
  }
]
```

```

        ],
    },
    "minOccurs": "1",
    "maxOccurs": "1"
},
{
    "id": "red_factor",
    "title": "the factor of the NRI",
    "input": {
        "literalDataDomains": [
            {
                "valueDefinition": {
                    "anyValue": true
                },
                "dataType": {
                    "name": "number"
                }
            }
        ]
    },
    "minOccurs": "1",
    "maxOccurs": "1",
    "additionalParameters": [
        {
            "role": "parameters": [
                {
                    "name": "ValueConstraints",
                    "min": 0.0,
                    "max": 1.0
                }
            ]
        }
    ]
},
{
    "id": "nir_factor",
    "title": "the factor of the NIR",
    "input": {
        "literalDataDomains": [
            {
                "valueDefinition": {
                    "anyValue": true
                },
                "dataType": {
                    "name": "number"
                }
            }
        ]
    }
}

```

```

        },
        "minOccurs": "1",
        "maxOccurs": "1",
        "additionalParameters": [
            {
                "role": "parameters": [
                    {
                        "name": "ValueConstraints",
                        "min": 0.0,
                        "max": 1.0
                    }
                ]
            }
        ],
        "outputs": [
            {
                "id": "raster",
                "title": "The raster of the resulting
NDVI calculation",
                "input": {
                    "formats": [
                        {
                            "mimeType": "image/tiff",
                            "default": true
                        }
                    ]
                },
                "additionalParameters": [
                    {
                        "role": "parameters": [
                            {
                                "_comment": "The
Profile element allows to define a more specific format than the registered MIME Media
Type (e.g. GeoTiff)",
                                "name": "Profile",
                                "values": [
                                    "http://www.opengis.net/eoc/formats/raster/GeoTiff"
                                ]
                            }
                        ]
                    }
                ]
            }
        ]
    }
}

```

```
        ],
      },
      "processVersion": "1.0.0",
      "jobControlOptions": [
        "async-execute"
      ],
      "outputTransmission": [
        "reference"
      ]
    }
  }
}
```

JSON-LD (Expanded)

```
{ "@context": {  
    "@version": "1.1",  
    "dct": "http://purl.org/dc/terms/",  
    "atom": "http://www.w3.org/2005/Atom/",  
    "iana": "http://www.iana.org/assignments/relation/",  
    "os": "http://a9.com/-/spec/opensearch/1.1/",  
    "eop": "http://www.opengis.net/ont/eo-geojson/1.0/",  
    "owc": "http://www.opengis.net/ont/owc/1.0/",  
    "gj": "https://purl.org/geojson/vocab#",  
    "gsp": "http://www.opengis.net/ont/geosparql#",  
    "vcard": "http://www.w3.org/2006/vcard/ns#",  
    "skos": "http://www.w3.org/2004/02/skos/core#",  
    "dcat": "http://www.w3.org/ns/dcat#",  
    "xsd": "http://www.w3.org/2001/XMLSchema#",  
    "prov": "http://www.w3.org/ns/prov#",  
    "locn": "http://www.w3.org/ns/locn#",  
    "foaf": "http://xmlns.com/foaf/0.1/",  
    "wps": "http://www.opengis.net/wps/2.0/",  
    "schema": "http://schema.org/",  
    "adms": "http://www.w3.org/ns/adms#",  
    "owl": "http://www.w3.org/2002/07/owl#",  
    "rdfs": "http://www.w3.org/2000/01/rdf-schema#" } }
```

```

},
"@id": "http://fedeo.esa.int/services/NdviCalculation",
"@type": "dcat:DataService",
"dct:description": "Calculation of NDVI using the SNAP toolbox for Sentinel-2",
"dct:identifier": "NdviCalculation",
"dct:language": {
  "@id": "http://id.loc.gov/vocabulary/iso639-1/en"
},
"dct:modified": "2019-05-20T00:00:00Z",
"dct:title": "Calculation of NDVI using the SNAP toolbox for Sentinel-2",
"dct:type": "http://inspire.ec.europa.eu/metadata-codelist/ResourceType/service",
"owc:links": {
  "iana:alternate": {
    "@id":
    "http://fedeo.esa.int/services/NdviCalculation?httpAccept=application/vnd.iso.19139-2%2Bxml",
    "dct:title": "ISO 19139-2 metadata",
    "atom:type": "application/vnd.iso.19139-2+xml"
  },
  "iana:describedby": {
    "@id": "http://fedeo.esa.int/services/NdviCalculation?httpAccept=text/html",
    "dct:title": "Described by",
    "atom:type": "text/html"
  },
  "iana:profile": [
    {
      "@id": "http://www.opengis.net/spec/owc-geojson/1.0/req/core"
    },
    {
      "@id": "http://www.opengis.net/spec/eopad-geojson/1.0/req/core"
    }
  ]
},
"owc:offerings": [
{
  "@type": "owc:Offering",
  "owc:code": "http://www.opengis.net/spec/owc-geojson/1.0/req/docker/image",
  "owc:contents": {
    "owc:content": "docker.52north.org/eopad/snap-sentinel-2-ndvi:latest",
    "owc:type": "text/plain"
  }
},
{
  "@type": "owc:Offering",
  "owc:code": "http://www.opengis.net/spec/owc-geojson/1.0/req/wps",
  "owc:operations": {
    "owc:code": "DescribeProcess",
    "owc:href": "http://52north.org/eopad/wps/processes/NdviCalculation",
    "owc:method": "GET",
    "owc:result": {
      "owc:content": {
        "owc:content": "{'process': 'NdviCalculation', 'parameters': {'input_raster': 'Sentinel-2', 'output_ndvi': 'Ndvi'} }"
      }
    }
  }
}
]
}

```

```

    "wps:processDescription": {
        "wps:process": {
            "@id": "NdviCalculation"
        }
    },
    "owc:type": "application/json"
},
"owc:type": "application/json"
}
}
],
"owl:versionInfo": "0.1",
"adms:versionNotes": "Initial version of application for use in OGC Testbed-15.",
"dcat:contactPoint": {
    "@type": "vcard:Organization",
    "vcard:hasName": "52 North"
},
"dcat:keyword": "Vegetation"
}

```

C.2.3. Example 3: Cloud coverage for an area of interest

Application Package (DeployProcess)

```

{
  "processDescription": {
    "process": {
      "id": "org.n52.project.tb15.eopad.SentinelQualityAreaOfInterest",
      "title": "Calculation of quality of a area of interest for Sentinel-2",
      "description": "Calculation of quality of a area of interest for Sentinel-2",
      "version": "1.0.0",
      "keywords": ["Cloud coverage"],
      "inputs": [
        {
          "id": "source",
          "title": "Source Input Image (sentinel-2 product in SAFE format)",
          "description": "Source Input Image (sentinel-2 product in SAFE format).
Either provided inline, by reference or by product ID",
          "minOccurs": 1,
          "maxOccurs": 1,
          "input": {
            "formats": [
              {
                "mimeType": "text/plain",
                "default": true
              },
              {
                "mimeType": "application/zip",
                "default": false
              }
            ]
          }
        }
      ]
    }
  }
}

```

```

        }
    ]
}
},
{
    "id": "area_of_interest",
    "title": "Area of Interest to be considered during the quality assurance",
    "description": "Area of Interest to be considered during the quality assurance (e.g. cloud coverage will only play a role in the specific polygon)",
    "minOccurs": 1,
    "maxOccurs": 1,
    "input": {
        "formats": [
            {
                "mimeType": "application/geo+json",
                "default": true
            }
        ]
    }
},
"outputs": [
{
    "id": "cloud_coverage",
    "title": "The cloud coverage for the selected area of interest.",
    "description": "The cloud coverage for the selected area of interest.",
    "output": {
        "literalDataDomains": [
            {
                "default": true,
                "dataType": { "name": "float" },
                "valueDefinition": [
                    {
                        "minimumValue": "0.0",
                        "maximumValue": "100.0",
                        "rangeClosure": "closed"
                    }
                ]
            }
        ]
    }
},
"jobControlOptions": ["async-execute"],
"outputTransmission": ["reference"]
},
"immediateDeployment": true,
"executionUnit": [
{
    "unit": {

```

```
        "type": "docker",
        "image": "52north/testbed-eopad-quality:latest"
    }
}
],
"deploymentProfileName": "http://www.opengis.net/profiles/eoc/dockerizedApplication"
}
```

Appendix D: JSON Schema Documents

The [JSON Schema](#) representation (Draft 4) is used to define the schemas in this annex. OpenAPI uses an extended subset of JSON Schema Draft 5 which is described at <https://swagger.io/docs/specification/data-models/keywords/>.

The JSON Schema Documents are named:

```
eopad-geojson-schema.json  
facetedResults-schema.json  
explain-schema.json
```

All these JSON Schema Documents contain documentation of the meaning of each element and attribute, and this documentation shall be considered normative as specified in Subclause 11.6.3 of [OGC 06-121r9].

D.1. Schemas for EOPAD JSON validation

eopad-geojson-schema.json

```
{  
  "$schema": "http://json-schema.org/draft-04/schema#",  
  "title": "GeoJSON encoding of EO Processing and Application metadata",  
  "description": "Definition of document with EO Processing and Application metadata  
in GeoJSON encoding. Note that numbers in the instance should not be surrounded by  
double-quotes to validate against this schema.",  
  "$ref": "#/definitions/Feature",  
  "definitions": {  
    "Point": {  
      "title": "Point",  
      "type": "object",  
      "properties": {  
        "coordinates": {  
          "title": "coordinates",  
          "description": "One position (longitude, latitude)",  
          "type": "array",  
          "minItems": 2,  
          "maxItems": 2,  
          "items": {  
            "type": "number"  
          }  
        },  
        "type": {  
          "type": "string",  
          "enum": [  
            "Point"  
          ]  
        }  
      }  
    },  
    "type": {  
      "type": "string",  
      "enum": [  
        "Point"  
      ]  
    }  
  }  
}
```

```

},
  "required": [
    "coordinates",
    "type"
  ],
  "additionalProperties": false
},
"MultiPoint": {
  "title": "MultiPoint",
  "type": "object",
  "properties": {
    "coordinates": {
      "title": "coordinates",
      "description": "Array of positions",
      "type": "array",
      "minItems": 1,
      "items": {
        "description": "One position",
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": {
          "type": "number"
        }
      }
    },
    "additionalItems": false
  },
  "type": {
    "type": "string",
    "enum": [
      "MultiPoint"
    ]
  }
},
"required": [
  "coordinates",
  "type"
],
"additionalProperties": false
},
"LineString": {
  "title": "LineString",
  "type": "object",
  "properties": {
    "coordinates": {
      "title": "coordinates",
      "description": "Array of positions",
      "type": "array",
      "minItems": 2,

```

```

    "items": {
        "description": "One position",
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": {
            "type": "number"
        }
    }
},
"type": {
    "type": "string",
    "enum": [
        "LineString"
    ]
}
},
"required": [
    "coordinates",
    "type"
],
"additionalProperties": false
},
"MultiLineString": {
    "title": "MultiLineString",
    "type": "object",
    "properties": {
        "coordinates": {
            "title": "coordinates",
            "description": "Array of linestring",
            "type": "array",
            "minItems": 1,
            "items": {
                "description": "Linestring, i.e. array of positions",
                "type": "array",
                "minItems": 2,
                "maxItems": 2,
                "items": {
                    "type": "number"
                }
            }
        }
    }
},
"type": {
    "type": "string",
    "enum": [
        "MultiLineString"
    ]
}
}

```

```

        ]
    }
},
"required": [
    "coordinates",
    "type"
],
"additionalProperties": false
},
"Polygon": {
    "title": "Polygon",
    "type": "object",
    "properties": {
        "coordinates": {
            "title": "coordinates",
            "description": "Array of linestrings",
            "type": "array",
            "minItems": 1,
            "items": {
                "description": "Linear ring, i.e. linestring or array of
positions",
                "type": "array",
                "minItems": 1,
                "items": {
                    "description": "One position",
                    "type": "array",
                    "minItems": 2,
                    "maxItems": 2,
                    "items": {
                        "type": "number"
                    }
                }
            }
        },
        "type": {
            "type": "string",
            "enum": [
                "Polygon"
            ]
        }
    },
    "required": [
        "coordinates",
        "type"
    ],
    "additionalProperties": false
},
"MultiPolygon": {
    "title": "MultiPolygon",
    "type": "object",
    "properties": {

```

```

"coordinates": {
    "title": "coordinates",
    "description": "Array of Polygons",
    "type": "array",
    "minItems": 1,
    "items": {
        "description": "Array of linestrings",
        "type": "array",
        "minItems": 1,
        "items": {
            "description": "Linear ring, i.e. linestring or array of
positions",
            "type": "array",
            "items": {
                "description": "One position",
                "type": "array",
                "minItems": 2,
                "maxItems": 2,
                "items": {
                    "type": "number"
                }
            }
        }
    }
},
"type": {
    "type": "string",
    "enum": [
        "MultiPolygon"
    ]
}
},
"required": [
    "coordinates",
    "type"
],
"additionalProperties": false
},
"Geometry": {
    "title": "Geometry",
    "type": "object",
    "oneOf": [
        {
            "$ref": "#/definitions/Point"
        },
        {
            "$ref": "#/definitions/MultiPoint"
        },
        {
            "$ref": "#/definitions/LineString"
        }
    ]
}

```

```

    {
        "$ref": "#/definitions/MultiLineString"
    },
    {
        "$ref": "#/definitions/Polygon"
    },
    {
        "$ref": "#/definitions/MultiPolygon"
    }
]
},
"Links": {
    "title": "Links",
    "description": "OGC 14-055r2",
    "type": "object",
    "properties": {
        "type": {
            "type": "string",
            "enum": [
                "Links"
            ]
        }
    },
    "patternProperties": {
        "^(profiles|alternates|via|previews)": {
            "description": "OGC 14-055r2",
            "type": "array",
            "minItems": 1,
            "items": {
                "$ref": "#/definitions/Link"
            }
        },
        "^(search|describedby|related)": {
            "type": "array",
            "minItems": 1,
            "items": {
                "$ref": "#/definitions/Link"
            }
        }
    },
    "additionalProperties": {
        "type": "array",
        "minItems": 1,
        "items": {
            "$ref": "#/definitions/Link"
        }
    }
},
"Properties": {
    "title": "Properties",
    "type": "object",

```

```

"allOf": [
    {
        "type": "object",
        "properties": {
            "acquisitionInformation": {
                "description": "OGC 17-003",
                "type": "array",
                "items": {
                    "$ref": "#/definitions/AcquisitionInformation"
                }
            },
            "productInformation": {
                "$ref": "#/definitions/ProductInformation"
            },
            "spatial": {
                "$ref": "#/definitions/Location"
            },
            "priceSpecification": {
                "$ref": "#/definitions/CompoundPriceSpecification"
            }
        }
    },
    {
        "$ref": "#/definitions/MetadataInformation"
    },
    {
        "$ref": "#/definitions/ServiceIdentification"
    },
    {
        "$ref": "#/definitions/RightsInformation"
    },
    {
        "$ref": "#/definitions/DescriptiveKeywords"
    },
    {
        "$ref": "#/definitions/RelatedUrl"
    }
],
},
"Feature": {
    "description": "GeoJSON Feature",
    "type": "object",
    "properties": {
        "@context": {},
        "type": {
            "title": "type",
            "type": "string",
            "enum": [
                "Feature"
            ]
        }
    }
}
]
}

```

```
"id": {
    "type": "string",
    "format": "uri"
},
"geometry": {
    "oneOf": [
        {
            "$ref": "#/definitions/Geometry"
        },
        {
            "type": "null"
        }
    ]
},
"properties": {
    "$ref": "#/definitions/Properties"
},
"bbox": {
    "type": "array",
    "minItems": 4,
    "maxItems": 4,
    "items": {
        "type": "number"
    }
}
},
"required": [
    "type",
    "id",
    "geometry",
    "properties"
]
},
"Link": {
    "description": "OGC 14-055r2",
    "type": "object",
    "properties": {
        "href": {
            "type": "string",
            "format": "uri"
        },
        "type": {
            "description": "MIME type",
            "type": "string"
        },
        "title": {
            "type": "string"
        },
        "length": {
            "type": "integer",
            "minimum": 0
        }
    }
}
```

```

        "exclusiveMinimum": true
    },
    "lang": {
        "description": "RFC-3066",
        "type": "string"
    }
},
"required": [
    "href"
]
},
"Offering": {
    "title": "Offering",
    "description": "Offering as defined in OGC 14-055r2",
    "type": "object",
    "properties": {
        "code": {
            "type": "string",
            "format": "uri"
        },
        "operations": {
            "type": "array",
            "items": {
                "$ref": "#/definitions/Operation"
            }
        },
        "contents": {
            "type": "array"
        },
        "styles": {
            "type": "array"
        }
    },
    "required": [
        "code"
    ]
},
"Operation": {
    "description": "OGC 14-055r2",
    "type": "object",
    "properties": {
        "code": {
            "type": "string"
        },
        "method": {
            "type": "string",
            "enum": [
                "GET",
                "POST",
                "PUT",
                "HEAD",
                "DELETE"
            ]
        }
    }
}

```

```

        "PATCH",
        "DELETE"
    ],
},
"type": {
    "description": "Media type",
    "type": "string"
},
"href": {
    "type": "string",
    "format": "uri"
},
"request": {
    "type": "object"
},
"result": {
    "type": "object"
}
},
"required": [
    "code",
    "method",
    "href"
]
},
"Agent": {
    "description": "vcard:Kind, foaf:Agent subset. OGC 14-055r2.",
    "type": "object",
    "minProperties": 1,
    "properties": {
        "type": {
            "type": "string",
            "enum": [
                "Individual",
                "Kind",
                "Person",
                "Organization",
                "Agent"
            ]
        },
        "name": {
            "type": "string"
        },
        "email": {
            "type": "string",
            "format": "email"
        },
        "uri": {
            "type": "string",
            "format": "uri"
        }
    }
}

```

```

    }
},
"Category": {
  "description": "OGC 14-055r2 §7.1.1.15",
  "type": "object",
  "properties": {
    "type": {
      "type": "string",
      "enum": [
        "Category"
      ]
    },
    "scheme": {
      "type": "string",
      "format": "uri"
    },
    "term": {
      "type": "string"
    },
    "label": {
      "type": "string"
    }
  },
  "required": [
    "term"
  ],
  "additionalProperties": false
},
"DescriptiveKeywords": {
  "type": "object",
  "properties": {
    "subject": {
      "description": "Topic category",
      "type": "array",
      "minItems": 1,
      "items": {
        "$ref": "#/definitions/Category"
      }
    },
    "categories": {
      "description": "OGC 14-055r2 - Keywords associated with a controlled vocabulary.",
      "type": "array",
      "minItems": 1,
      "items": {
        "$ref": "#/definitions/Category"
      }
    },
    "keyword": {
      "description": "dcat:keyword. Keywords not associated with a controlled vocabulary."
    }
  }
}

```

```

        "type": "array",
        "minItems": 1,
        "items": {
            "type": "string",
            "minLength": 1
        }
    },
    "theme": {
        "description": "dcat:theme",
        "type": "array",
        "minItems": 1,
        "items": {
            "$ref": "#/definitions/Concept"
        }
    }
},
"ServiceContact": {
    "type": "object",
    "properties": {
        "publisher": {
            "description": "role=\"publisher\" (OGC 14-055r2)",
            "type": "string"
        },
        "authors": {
            "description": "role=\"creator\" (OGC 14-055r2)",
            "type": "array",
            "minItems": 1,
            "items": {
                "$ref": "#/definitions/Agent"
            }
        },
        "contactPoint": {
            "description": "Role=\"contact point\"",
            "type": "array",
            "minItems": 1,
            "items": {
                "$ref": "#/definitions/Agent"
            }
        },
        "qualifiedAttribution": {
            "description": "Role=other",
            "type": "array",
            "minItems": 1,
            "items": {
                "$ref": "#/definitions/Attribution"
            }
        }
    }
},
"MetadataInformation": {

```

```

"type": "object",
"properties": {
    "updated": {
        "description": "OGC 14-055r2",
        "type": "string",
        "format": "date-time"
    },
    "lang": {
        "description": "OGC 14-055r2",
        "type": "string"
    },
    "published": {
        "type": "string",
        "format": "date-time"
    }
},
"required": [
    "updated"
]
},
"ServiceIdentification": {
    "type": "object",
    "allOf": [
        {
            "type": "object",
            "properties": {
                "kind": {
                    "description": "dct:type",
                    "type": "string",
                    "format": "uri"
                },
                "title": {
                    "description": "OGC 14-055r2",
                    "type": "string"
                },
                "identifier": {
                    "type": "string"
                },
                "date": {
                    "description": "OGC 14-055r2",
                    "type": "string"
                },
                "abstract": {
                    "description": "OGC 14-055r2",
                    "type": "string"
                },
                "provenance": {
                    "description": "dct:ProvenanceStatement",
                    "type": "array",
                    "minItems": 1,
                    "items": {
                        "type": "object"
                    }
                }
            }
        }
    ]
}

```

```

        "$ref": "#/definitions/ProvenanceStatement"
    }
},
"wasUsedBy": {
    "description": "prov:Activity",
    "type": "array",
    "minItems": 1,
    "items": {
        "$ref": "#/definitions/Activity"
    }
},
"doi": {
    "description": "adms:identifier",
    "type": "string"
},
"bibliographicCitation": {
    "description": "dct:bibliographicCitation",
    "type": "string"
},
"versionInfo": {
    "description": "owl:versionInfo",
    "type": "string"
},
"versionNotes": {
    "description": "adms:versionNotes",
    "type": "string"
},
"required": [
    "title",
    "identifier"
],
{
    "$ref": "#/definitions/ServiceContact"
}
],
},
"Attribution": {
    "description": "prov:Attribution",
    "type": "object",
    "properties": {
        "type": {
            "type": "string",
            "enum": [
                "Attribution"
            ]
        },
        "role": {
            "description": "ResponsibleParty role",
            "type": "string",

```

```

    "enum": [
        "resourceProvider",
        "custodian",
        "owner",
        "user",
        "distributor",
        "originator",
        "pointOfContact",
        "principalInvestigator",
        "processor",
        "publisher",
        "author"
    ],
},
"agent": {
    "type": "array",
    "items": {
        "$ref": "#/definitions/Agent"
    }
},
},
"required": [
    "role",
    "agent"
],
},
"AcquisitionParameters": {
    "type": "object",
    "allOf": [
        {
            "$ref": "#/definitions/TemporalInformation"
        }
    ]
},
"TemporalInformation": {
    "description": "Subset of AcquisitionParameters",
    "type": "object",
    "properties": {
        "beginningDateTime": {
            "type": "string",
            "format": "date-time"
        },
        "endingDateTime": {
            "type": "string",
            "format": "date-time"
        }
    },
    "required": [
        "beginningDateTime"
    ]
},

```

```
"AcquisitionInformation": {
    "type": "object",
    "properties": {
        "type": {
            "type": "string",
            "enum": [
                "AcquisitionInformation"
            ]
        },
        "platform": {
            "$ref": "#/definitions/Platform"
        },
        "instrument": {
            "$ref": "#/definitions/Instrument"
        }
    }
},
"Platform": {
    "type": "object",
    "minProperties": 1,
    "properties": {
        "type": {
            "type": "string",
            "enum": [
                "Platform"
            ]
        },
        "id": {
            "type": "string",
            "format": "uri"
        },
        "platformShortName": {
            "type": "string"
        },
        "platformSerialIdentifier": {
            "type": "string"
        },
        "orbitType": {
            "type": "string",
            "enum": [
                "GEO",
                "LEO"
            ]
        }
    },
    "required": [
        "platformShortName"
    ],
    "additionalProperties": false
},
"Instrument": {
```

```

"type": "object",
"properties": {
    "type": {
        "type": "string",
        "enum": [
            "Instrument"
        ]
    },
    "id": {
        "type": "string",
        "format": "uri"
    },
    "sensorType": {
        "type": "string",
        "enum": [
            "OPTICAL",
            "RADAR",
            "ATMOSPHERIC",
            "ALTIMETRIC",
            "LIMB"
        ]
    },
    "instrumentShortName": {
        "type": "string"
    },
    "description": {
        "type": "string"
    }
},
"required": [
    "instrumentShortName"
],
"additionalProperties": false
},
"ProductInformation": {
    "description": "OGC 17-003",
    "type": "object"
},
"RelatedUrl": {
    "type": "object",
    "properties": {
        "links": {
            "description": "OGC 14-055r2 and OGC 17-069r2",
            "oneOf": [
                {
                    "$ref": "#/definitions/Links"
                },
                {
                    "description": "OGC 17-069r2",
                    "type": "array",
                    "items": {
                        "type": "string"
                    }
                }
            ]
        }
    }
}

```

```

                "$ref": "#/definitions/Link_"
            }
        }
    ]
},
"offerings": {
    "description": "OGC 14-055r2",
    "type": "array",
    "minItems": 1,
    "items": {
        "$ref": "#/definitions/Offering"
    }
},
"endpointDescription": {
    "description": "dcat:endpointDescription",
    "type": "array"
},
"hasPart": {
    "description": "dct:hasPart (coupled resource).",
    "type": "array",
    "minItems": 1,
    "items": {
        "$ref": "#/definitions/CoupledResource"
    }
}
},
"Location": {
    "description": "Alternative representation of geometry.",
    "type": "object",
    "properties": {
        "type": {
            "type": "string",
            "enum": [
                "Location"
            ]
        },
        "id": {
            "type": "string",
            "format": "uri"
        }
    },
    "geometry": {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {
                "type": {
                    "type": "string"
                },
                "value": {
                    "type": "string"
                }
            }
        }
    }
}

```

```

        }
    }
}
},
"LicenseDocument": {
    "description": "dct:LicenseDocument",
    "type": "object",
    "properties": {
        "type": {
            "type": "string",
            "enum": [
                "LicenseDocument"
            ]
        },
        "label": {
            "description": "rdfs:label",
            "type": "string"
        }
    },
    "required": [
        "label"
    ]
},
"RightsStatement": {
    "description": "dct:RightsStatement",
    "type": "object",
    "properties": {
        "type": {
            "type": "string",
            "enum": [
                "RightsStatement"
            ]
        },
        "label": {
            "description": "rdfs:label",
            "type": "string"
        }
    },
    "required": [
        "label"
    ]
},
"ProvenanceStatement": {
    "title": "dct:provenance",
    "description": "dct:ProvenanceStatement",
    "type": "object",
    "properties": {
        "type": {
            "type": "string",

```

```

    "enum": [
        "ProvenanceStatement"
    ],
    "label": {
        "description": "rdfs:label",
        "type": "string"
    },
    "required": [
        "label"
    ],
},
"Activity": {
    "description": "prov:Activity",
    "type": "object",
    "properties": {
        "type": {
            "type": {
                "type": "string",
                "enum": [
                    "Activity"
                ]
            },
            "generated": {
                "$ref": "#/definitions/Entity"
            },
            "qualifiedAssociation": {
                "$ref": "#/definitions/Association"
            }
        },
        "required": [
            "generated",
            "qualifiedAssociation"
        ]
    },
},
"Entity": {
    "description": "prov:Entity",
    "type": "object",
    "properties": {
        "type": {
            "type": "string",
            "enum": [
                "Entity"
            ]
        },
        "degree": {
            "type": "string",
            "format": "uri"
        },
        "description": {
            "type": "string"
        }
    }
}

```

```

        }
    },
    "required": [
        "degree"
    ]
},
"Association": {
    "description": "prov:Association",
    "type": "object",
    "properties": {
        "type": {
            "type": "string",
            "enum": [
                "Association"
            ]
        },
        "hadPlan": {
            "$ref": "#/definitions/Plan"
        }
    },
    "required": [
        "hadPlan"
    ]
},
"Plan": {
    "description": "prov:Plan",
    "type": "object",
    "properties": {
        "type": {
            "type": "string",
            "enum": [
                "Plan"
            ]
        },
        "wasDerivedFrom": {
            "$ref": "#/definitions/Standard"
        }
    },
    "required": [
        "wasDerivedFrom"
    ]
},
"Standard": {
    "description": "dct:Standard",
    "type": "object",
    "properties": {
        "type": {
            "type": "string",
            "enum": [
                "Standard"
            ]
        }
    }
}

```

```

        },
        "title": {
            "type": "string"
        },
        "issued": {
            "description": "dct:issued",
            "type": "string",
            "format": "date-time"
        }
    },
    "required": [
        "title"
    ]
},
"Concept": {
    "description": "skos:Concept",
    "type": "object",
    "properties": {
        "id": {
            "type": "string",
            "format": "uri"
        }
    }
},
"CoupledResource": {
    "description": "GeoDCAT-AP §II.6 - coupled resource",
    "type": "string",
    "format": "uri"
},
"RightsInformation": {
    "type": "object",
    "properties": {
        "rights": {
            "description": "dct:rights and OGC 14-055r2 §7.1.2.7",
            "type": "string"
        },
        "license": {
            "description": "dct:LicenseDocument",
            "type": "array",
            "items": {
                "$ref": "#/definitions/LicenseDocument"
            }
        },
        "accessRights": {
            "description": "dct:RightsStatement",
            "type": "array",
            "items": {
                "$ref": "#/definitions/RightsStatement"
            }
        }
    }
}

```

```

},
"Link_": {
  "description": "OGC 17-069r2",
  "type": "object",
  "properties": {
    "href": {
      "type": "string",
      "format": "uri"
    },
    "rel": {
      "type": "string"
    },
    "type": {
      "description": "MIME type",
      "type": "string"
    },
    "hreflang": {
      "description": "RFC-3066",
      "type": "string"
    },
    "title": {
      "type": "string"
    },
    "length": {
      "type": "integer",
      "minimum": 0,
      "exclusiveMinimum": true
    }
  },
  "required": [
    "href"
  ]
},
"CompoundPriceSpecification": {
  "description": "schema.org",
  "type": "object",
  "properties": {
    "type": {
      "type": {
        "type": "string",
        "enum": [
          "CompoundPriceSpecification"
        ]
      }
    },
    "description": {
      "type": "string"
    },
    "priceComponent": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/UnitPriceSpecification"
      }
    }
  }
}

```

```

        }
    },
    "UnitPriceSpecification": {
        "description": "schema.org",
        "type": "object"
    }
}
}

```

D.2. Schemas for Faceted Results validation

The JSON Schemas included in OGC 17-047 can be used to validate the OpenSearch response (i.e. FeatureCollection) in GeoJSON format. The facetResults property presented in chapter 10 ‘Future Work’ of OGC 17-047 is defined in the JSON schema fragments presented below and consistent with the OASIS searchRetrieve facetResults XML schema available at <http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/os/schemas/facetedResults.xsd>.

Testbed-15 proposes that the OGC 17-047 JSON schemas are extended with the schema elements below.

facetResults-schema.json

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "title": "GeoJSON encoding of optional FacetedResults object in OpenSearch Response as per Chapter 10 of OGC 17-047.",
    "description": "Definition of the FacetedResults object with OpenSearch Response in GeoJSON encoding. The definition is the JSON equivalent of the OASIS searchRetrieve facet schema available at http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/os/schemas/facetedResults.xsd. Note that numbers in the instance should not be surrounded by double-quotes to validate against this schema. ",
    "$ref": "#/definitions/FacetedResults",
    "definitions": {

```

D.2.1. Definitions/Query (OGC 17-047)

```

    "Query": {
        "description": "OpenSearch Query element as defined at
http://www.opensearch.org/Specifications/OpenSearch/1.1#OpenSearch\_Query\_element.",
        "type": "object",
        "properties": {
            "type": {
                "type": "string",
                "enum": [
                    "Query"
                ]
            },

```

```

    "title": {
        "type": "string"
    },
    "totalResults": {
        "type": "integer",
        "minimum": 0
    },
    "searchTerms": {
        "type": "string"
    },
    "count": {
        "type": "integer",
        "minimum": 0
    },
    "startIndex": {
        "type": "integer",
        "minimum": 0
    },
    "startPage": {
        "type": "integer",
        "minimum": 0
    },
    "language": {
        "type": "string"
    },
    "inputEncoding": {
        "type": "string"
    },
    "outputEncoding": {
        "type": "string"
    }
},
"additionalProperties": {
    "title": "Search parameter",
    "description": "Remaining OpenSearch query parameters and their values.",
    "type": [
        "string",
        "number"
    ]
},
"Term": {
    "type": "object",
    "properties": {
        "actualTerm": {
            "type": "string"
        },
        "query": {
            "$ref": "#/definitions/Query"
        }
    }
}

```

```

    "requestUrl": {
        "type": "string",
        "format": "uri"
    },
    "count": {
        "type": "integer"
    }
},
"required": [
    "actualTerm",
    "count"
],
"additionalProperties": false
},

```

D.2.2. Definitions/Facet

```

"Facet": {
    "type": "object",
    "properties": {
        "displayLabel": {
            "type": "string"
        },
        "description": {
            "type": "string"
        },
        "index": {
            "description": "OpenSearch queryable, e.g. eo:platform.",
            "type": "string"
        },
        "relation": {
            "description": "Default is \"=\\\".",
            "type": "string"
        },
        "terms": {
            "title": "term",
            "type": "array",
            "items": {
                "$ref": "#/definitions/Term"
            }
        }
    },
    "required": [
        "terms"
    ],
    "additionalProperties": false
},

```

D.2.3. Definitions/DataSource

```
"DataSource": {  
    "type": "object",  
    "properties": {  
        "displayLabel": {  
            "type": "string"  
        },  
        "description": {  
            "type": "string"  
        },  
        "baseURL": {  
            "type": "string",  
            "format": "uri"  
        },  
        "facets": {  
            "title": "facet",  
            "type": "array",  
            "items": {  
                "$ref": "#/definitions/Facet"  
            }  
        },  
        "required": [  
            "facets"  
        ],  
        "additionalProperties": false  
    },
```

D.2.4. Definitions/FacetedResults

```

"FacetedResults": {
    "title": "facetedResults",
    "type": "object",
    "oneOf": [
        {
            "type": "object",
            "properties": {
                "datasource": {
                    "type": "array",
                    "minItems": 1,
                    "items": {
                        "$ref": "#/definitions/DataSource"
                    }
                }
            },
            "additionalProperties": false
        },
        {
            "type": "object",
            "properties": {
                "facets": {
                    "type": "array",
                    "minItems": 1,
                    "items": {
                        "$ref": "#/definitions/Facet"
                    }
                }
            },
            "additionalProperties": false
        }
    ]
}
}

```

D.3. Schemas for Explain Document validation

explain-schema.json

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "title": "JSON Schema encoding of OASIS SearchRetrieve Explain Document",
    "description": "Only the subset of the Explain document required to discover facets is encoded. Complete modelling of the original XML Schema available at http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/os/schemas/explain.xsd is out of scope of the OGC Testbed-15 EOPAD activities and is left as future work. Additional information is available at http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/searchRetrieve-v1.0-part7-explain.html.",
    "$ref": "#/definitions/Explain",
}
```

```

"definitions": {
  "Explain": {
    "type": "object",
    "properties": {
      "serverInfo": {
        "$ref": "#/definitions/ServerInfo"
      },
      "indexInfo": {
        "type": "array",
        "items": {
          "$ref": "#/definitions/IndexInfo"
        }
      },
      "searchInfo": {
        "$ref": "#/definitions/SearchInfo"
      }
    },
    "additionalProperties": false
  },
  "ServerInfo": {
    "description": "basic details required to start a network connection to the described server",
    "type": "object",
    "properties": {
      "protocol": {
        "type": "string"
      },
      "version": {
        "type": "string"
      },
      "transport": {
        "type": "string"
      },
      "host": {
        "type": "string"
      },
      "port": {
        "type": "integer"
      },
      "database": {
        "type": "string"
      }
    },
    "required": [
      "host",
      "port",
      "database"
    ]
  },
  "IndexInfo": {
    "type": "object",

```

```

"properties": {
    "set": {
        "type": "array",
        "items": {
            "$ref": "#/definitions/Set"
        }
    },
    "index": {
        "type": "array",
        "items": {
            "$ref": "#/definitions/Index"
        }
    }
},
"additionalProperties": false
},
"Set": {
    "type": "object",
    "properties": {
        "name": {
            "description": "e.g. \"eo\"",
            "type": "string"
        },
        "identifier": {
            "description": "e.g. \"http://a9.com/-/opensearch/extensions/eo/1.0/\")",
            "type": "string",
            "format": "uri"
        }
    },
    "required": [
        "name",
        "identifier"
    ],
    "additionalProperties": false
},
"Index": {
    "type": "object",
    "properties": {
        "search": {
            "type": "boolean"
        },
        "sort": {
            "type": "boolean"
        },
        "map": {
            "type": "array",
            "items": {
                "$ref": "#/definitions/Map"
            }
        }
    }
}

```

```

    "configInfo": {
        "$ref": "#/definitions/ConfigInfo"
    }
},
"required": [
    "map"
]
},
"Map": {
    "type": "object",
    "properties": {
        "set": {
            "type": "string"
        },
        "name": {
            "type": "string"
        }
    },
    "required": [
        "set",
        "name"
    ],
    "additionalProperties": false
},
"ConfigInfo": {
    "type": "object",
    "properties": {
        "supports": {
            "description": "indicates that the server supports an element",
            "type": "array",
            "items": {
                "$ref": "#/definitions/StringPlusType"
            }
        }
    },
    "additionalProperties": false
},
"StringPlusType": {
    "type": "object",
    "properties": {
        "type": {
            "type": "string",
            "enum": [
                "value"
            ]
        },
        "value": {
            "type": "string"
        }
    },
    "required": [

```

```

        "type",
        "value"
    ],
    "additionalProperties": false
},
"SearchInfo": {
    "type": "object",
    "properties": {
        "facets": {
            "type": "array",
            "items": {
                "$ref": "#/definitions/Facet"
            }
        }
    }
},
"Facet": {
    "type": "object",
    "properties": {
        "facetType": {
            "description": "e.g. eo:organisationName",
            "type": "string"
        },
        "limit": {
            "$ref": "#/definitions/Limit"
        },
        "start": {
            "$ref": "#/definitions/Start"
        }
    },
    "required": [
        "facetType",
        "start"
    ]
},
"Limit": {
    "type": "object",
    "properties": {
        "limitDefault": {
            "type": "integer"
        },
        "otherLimit": {
            "type": "boolean"
        }
    },
    "required": [
        "limitDefault",
        "otherLimit"
    ]
},
"Start": {

```

```
"type": "object",
"properties": {
    "startDefault": {
        "type": "integer"
    },
    "otherStart": {
        "type": "boolean"
    }
},
"required": [
    "startDefault",
    "otherStart"
]
}
}
```

Appendix E: OGC API - Processes, Transactional Profile OpenAPI 3.0 specification

```
openapi: 3.0.2
servers:
  - description: SwaggerHub API Auto Mocking
    url: https://virtserver.swaggerhub.com/autermann/ogc-api-processes-transactional/1.0.0
  - description: Testbed-15 EOPAD server
    url: https://testbed.dev.52north.org/eopad/rest
info:
  title: Transactional extension of OGC API - Processes
  description: >
    An extension of [OGC API - Processes](https://app.swaggerhub.com/apis/geoprocessing/WPS/1.0-draft.2)
    with transactional operations that allow the deployment of
    [application packages](http://docs.opengeospatial.org/per/18-049r1.html).
  version: "1.0.0"
  contact:
    name: Christian Autermann
    email: c.autermann@52north.org
    url: https://52north.org
  license:
    name: Apache 2.0
    url: "http://www.apache.org/licenses/LICENSE-2.0.html"
tags:
  - name: Transactional
    description: Secured Admin-only calls
paths:
  "/processes":
    post:
      tags:
        - Transactional
      operationId: deployProcess
      summary: Deploys a process
      description: Deploys a new process.
      requestBody:
        description: The application package to deploy.
        content:
          application/json:
            schema:
              $ref: "#/components/schemas/ApplicationPackage"
      responses:
        "201":
          description: process created
          content:
```

```

application/json:
  schema:
    $ref: https://raw.githubusercontent.com/opengeospatial/wps-rest-
binding/master/core/openapi/schemas/process.yaml
  "400":
    $ref: "#/components/responses/InvalidOrUnsupportedProcess"
  "409":
    $ref: "#/components/responses/ProcessAlreadyExists"
  default:
    $ref: "#/components/responses/InternalServerError"

"/processes/{id}":
  parameters:
    - in: path
      name: id
      description: The id of a process
      required: true
      schema:
        type: string
        example: buffer

  put:
    tags:
      - Transactional
    operationId: updateProcess
    summary: Updates a process
    description: Updates an existing process.
    requestBody:
      description: The application package to deploy.
      content:
        application/json:
          schema:
            $ref: "#/components/schemas/ApplicationPackage"
    responses:
      "204":
        description: The process was updated.
      "400":
        $ref: "#/components/responses/InvalidOrUnsupportedProcess"
      "404":
        $ref: "#/components/responses/ProcessNotFound"
      default:
        $ref: "#/components/responses/InternalServerError"

  delete:
    tags:
      - Transactional
    operationId: deleteProcess
    summary: Deletes a process
    description: Deletes a process
    responses:
      "204":

```

```

    description: The process was deleted.
  "400":
    $ref: "#/components/responses/InvalidOrUnsupportedProcess"
  "404":
    $ref: "#/components/responses/ProcessNotFound"
  default:
    $ref: "#/components/responses/InternalServerError"

components:
responses:
  ProcessNotFound:
    description: "The process with id {id} does not exist."
    content:
      application/json:
        schema:
          $ref: "#/components/schemas/Exception"
  InvalidOrUnsupportedProcess:
    description: application package is invalid or unsupported
    content:
      application/json:
        schema:
          $ref: "#/components/schemas/Exception"
  ProcessAlreadyExists:
    description: an existing process already exists
    content:
      application/json:
        schema:
          $ref: https://raw.githubusercontent.com/opengeospatial/wps-rest-
binding/master/core/openapi/schemas/exception.yaml
  InternalServerError:
    description: An error occurred.
    content:
      application/json:
        schema:
          $ref: "#/components/schemas/Exception"
schemas:
  Exception:
    $ref: https://raw.githubusercontent.com/opengeospatial/wps-rest-
binding/master/core/openapi/schemas/exception.yaml
  Process:
    $ref: https://raw.githubusercontent.com/opengeospatial/wps-rest-
binding/master/core/openapi/schemas/process.yaml
  ProcessOffering:
    type: object
    required:
      - process
    properties:
      process:
        $ref: "#/components/schemas/Process"
      jobControlOptions:
        type: array

```

```

items:
  $ref: https://raw.githubusercontent.com/opengeospatial/wps-rest-
binding/master/core/openapi/schemas/jobControlOptions.yaml
  outputTransmission:
    type: array
    items:
      $ref: https://raw.githubusercontent.com/opengeospatial/wps-rest-
binding/master/core/openapi/schemas/transmissionMode.yaml
  ApplicationPackage:
    type: object
    required:
      - deploymentProfile
      - executionUnit
      - processDescription
    properties:
      immediateDeployment:
        type: boolean
      deploymentProfile:
        type: string
        format: uri
        example: http://www.opengis.net/profiles/eoc/dockerizedApplication
      executionUnit:
        type: array
        minItems: 1
        items:
          type: object
          example:
            type: docker
            image: 52north/eopad-ndvi:latest
      processDescription:
        $ref: "#/components/schemas/ProcessOffering"

```

Appendix F: Revision History

Table 49. Revision History

Date	Editor	Release	Primary clauses modified	Descriptions
October 9, 2019	Y. Coene	r0	All	Version released to the Catalog and Metadata DWG.
October 11, 2019	Y. Coene	r1	§8.3.2	Issue about media type with profile property removed from Lessons learned in GMU Catalog Server section.
October 11, 2019	Y. Coene	r1	§8.3.7	Use of HTTP Allow header field added as future work in DataBio Catalog Server section.
October 25, 2019	Y. Coene	r1	§8.3.7	Info about Basic Authentication and Swagger editor figure included.
Dec. 2, 2019	C. Reed	r1	Various	Minor edits to get doc ready for publication